Ministry of Economic Development
Manatū Ōhanga
Intellectual Property Office of New Zealand

PCT/NZ2004/000121

REC'D 15 JUL 2004
WIPO PCT

CERTIFICATE

This certificate is issued in support of an application for Patent registration in a country outside New Zealand pursuant to the Patents Act 1953 and the Regulations thereunder.

I hereby certify that annexed is a true copy of the Provisional Specification as filed on 13 June 2003 with an application for Letters Patent number 526447 made by PAUL HANSEN and FRANZ OMBLER.

Dated 8 July 2004.

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

Neville Harris

Neville Harris
Commissioner of Patents, Trade Marks and Designs

COMMISSIONER OF PATENTS, TRADEMARKS & DESIGNS • NEW ZEALAND •

526447

NEW ZEALAND PATENTS ACT 1953

PROVISIONAL SPECIFICATION

A CHOICE-BASED METHOD FOR CALIBRATING ADDITIVE POINTS SYSTEM

We, PAUL HANSEN, a New Zealand citizen, of 24 Maitland Street, Dunedin, New Zealand, and FRANZ OMBLER, a New Zealand citizen, of 19 Edinburgh Terrace, Berhampore, Wellington, New Zealand, do hereby declare this invention to be described in the following statement:

INTELLECTUAL PROPERTY OFFICE OF N.Z.
13 JUN 2003
RECEIVED

A CHOICE-BASED METHOD FOR CALIBRATING ADDITIVE POINTS SYSTEMS

FIELD OF INVENTION

The invention relates to additive points systems and in particular to a system, method and computer program for determining (calibrating) the points values of additive points systems.

BACKGROUND TO THE INVENTION

Additive points systems (APSs), a type of multi-attribute utility model or multiple criteria decision analysis tool, are also known as 'linear', 'point-count' and 'scoring' systems. APSs are widely and increasingly used worldwide in most branches of medicine for prioritisation, diagnosis and predictive purposes and in a wide range of other applications, including selecting immigrants, assessing mortgage applications and predicting parole violations, business bankruptcies and college graduations. They are also used as a generic project appraisal tool.

Canada and New Zealand, for example, recently developed APSs for a wide range of elective surgeries and other publicly-funded health cares. APSs are also used by the immigration systems of New Zealand, Canada, Australia and Germany and, to a lesser extent, the United Kingdom.

APSs in general represent a relatively simple solution to the pervasive problem faced by decision makers with multiple criteria or attributes to consider, particularly when ranking alternatives or individuals. Hereinafter in this document alternatives and individuals are referred to generically as 'alternatives'.

Specifically, APSs serve to combine alternatives' characteristics on *multiple* criteria to produce a *single* ranking of alternatives with respect to an over-arching criterion (such as, for example, the order in which to treat patients), or, more simply, to reach a decision (for example, whether or not to admit an immigrant).

In addition to their having been near universally found in many studies to be more accurate than 'expert' decision makers in the respective fields to which they have been applied, the appeal of points systems is that they are simple to use.

Each alternative's categorical rating on each criterion that is deemed relevant to the over-arching criterion is scored a particular number of points, that usually increase with the 'importance' of the categories, and the points are *summed* (hence *additive* points systems) to produce a total score for the alternative.

Alternatives are ranked with respect to the over-arching criterion according to their total scores, including being declined altogether if a particular score 'threshold' is not reached. Usually the higher an alternative's score the higher its ranking, and the scores typically have no other meaning than this.

Most APSs for elective surgeries and immigration respectively (the examples referred to above), have between five and seven criteria and two to five categories on each criterion. Figures 1 and 2 show an example of an APS used in Canada for prioritising patients for hip or knee replacement surgery.

In this example there are seven criteria, mostly based on types of pain and functional limitations. Each criterion has a number of mutually exclusive and exhaustive categories on which the consulted decision maker (usually a doctor) is asked to rate the patient being considered. For example, item 2 in Figure 1, refers to the criterion of "Pain at Rest" and requires that the patient be assigned to one of four categories: "None", "Mild", "Moderate" or "Severe".

In general, if the criteria and the categories on each have been chosen for a particular APS system, then the point values for that APS must be determined (calibrated) such that the resulting ranking of alternatives represents the decision makers' preferences. The invention is a new system, method and computer program for doing this.

In addition to the arbitrary assignment of points, there are two main existing approaches to calibrating APSs.

The first regresses decision makers' judgements of the relative priorities or importance of a sample of real or hypothetical alternative 'profiles' (in other words, the alternative's categorical ratings on the criteria) on their characteristics in terms of the criteria and derives point values from the regression coefficients. Usually only a small proportion of existing or theoretically possible profiles is surveyed because of the responder burden on the consulted decision makers.

The above-mentioned decision makers' judgements are often elicited via a visual analogue scale (VAS). Item 8 of Figure 2 is such a VAS, where the decision maker is asked "to rate the urgency or relative priority of this patient" between "Not urgent at all" and "Extremely urgent (just short of an emergency)". This 'score' and the patient's characteristics in terms of Items 1 to 7 of the same figure, along with analogous data for other patients, may then be used to calibrate the point values for the various criteria and categories using multiple regression techniques, as explained above.

The second existing approach to calibration uses decision makers' judgements of the pairwise relative importance of the APS's criteria to derive ratio scale weights. These weights are then applied to normalised criteria values to derive point values. An example of this type of technique is the Analytic Hierarchical Process (AHP).

Thus the first approach assumes that decision makers' judgements have *interval scale* measurement properties and the second approach assumes they have *ratio scale* properties. Both assumptions are relatively stringent and current techniques for eliciting decision makers' judgements have well-known biases. With respect to the first approach, for example, the validity of the dependent variable (the experts' judgments) and therefore the point values derived from the estimated coefficients, can be criticised on two main grounds.

First, the scaling methods such as the VAS used to elicit the experts' judgments of the profiles' relative priorities are based on mere introspection rather than the expression of a choice. Second, VAS in general may not actually have the scaling measurement properties required for the valuations that they produce to be interpreted as relative priorities rather than just as rankings.

It is therefore desirable to have a method of calibrating new APSs or recalibrating or validating extant ones that requires only *ordinal* measurement properties, specifically the positive expression of a ranking over pairs of alternatives, as this is the least stringent of measurement property requirements. It would also be desirable for this method to achieve accurate results while reducing the burden on decision makers of ranking pairs of alternatives by minimising the number of pairs they have to rank.

SUMMARY OF THE INVENTION

In broad terms, in one form the invention provides a method for calibrating additive points systems (APSs) comprising a pre-determined plurality of criteria, each criterion having one or more categories wherein the points for each category of each criterion are determined by the pairwise ranking of profile pairs, each profile comprising a set of one or more of the criteria, each criterion in the set instantiated with one of the categories for that criterion.

Preferably the method comprises the steps of generating ambiguous profile pairs for the APS to be calibrated, resolving the ambiguous profile pairs, and solving the resulting system of equalities/inequalities to obtain the point values. Ambiguous profile pairs are profile pairs in which one profile has a *higher* categorical rating on at least one criterion and a *lower* rating on at least one other criterion than the other profile.

Preferably the step of generating the ambiguous profile pairs comprises the further step of removing any profiles that are theoretically impossible.

Preferably the step of generating the ambiguous profile pairs comprises the further step of reducing all profile pairs in which both profiles have one or more of the same criteria instantiated with the same category.

In broad terms in another form, the invention provides a system for calibrating additive points systems (APSs) comprising a pre-determined plurality of criteria for the additive points system, each criterion capable of being instantiated with one or more pre-defined categories; and a points calibrator configured to determine appropriate points for each category of each criterion by preparing data for and processing the results of the pairwise ranking of profile pairs, each profile comprising a set of one or more of the criteria, each criterion in the set instantiated with one of the categories for that criterion.

Preferably the points calibrator comprises an ambiguity generator configured to generate ambiguous profile pairs for the APS to be calibrated.

Preferably the points calibrator comprises a data input component configured to receive and store the equalities/inequalities that result from resolving the ambiguous profile pairs generated by the ambiguity generator.

Preferably the points calibrator comprises a solution component configured to solve the resulting system of equalities/inequalities to obtain the point values.

Preferably the ambiguity generator is further configured to remove any profiles that are theoretically impossible.

Preferably the ambiguity generator is further configured to reduce all profile pairs in which both profiles have one or more of the same criteria instantiated with the same category.

In broad terms in yet another form the invention provides a computer program for calibrating additive points systems (APSs) comprising an initialization component

configured to receive and store data representing a plurality of criteria for an APS and the categories with which each criterion may be instantiated, an ambiguity generator configured to generate ambiguous profile pairs for the APS to be calibrated, a resolution component configured to select and present profile pairs to a user to be explicitly resolved and to store the results of the resolution, an ambiguity management component configured to manage the resolved and unresolved ambiguities and to automatically resolve any ambiguities that can be resolved implicitly; and a solution component configured to solve the system of resolved inequalities/equalities from the resolution component and the ambiguity management component.

Preferably the computer program further comprises a revision component configured to allow a user to revise any resolved ambiguities.

Preferably the computer program is implemented using linear programming.

BRIEF DESCRIPTION OF THE FIGURES

Preferred forms of the method system and computer program for calibrating additive points systems will now be described with reference to the accompanying figures in which:

Figure 1 shows a prior art means of eliciting expert judgments for the purposes of calibrating an APS via multiple regression-based techniques;

Figure 2 shows a continuation of the prior art means of eliciting expert judgments for the purposes of calibrating an APS via multimedia regression-based techniques;

Figure 3 shows a preferred configuration of hardware for carrying out the invention;

Figure 4 shows a decision tree identifying the 12 rankings of the 8 profiles possible in an exemplar APS with three criteria and two categories and allowing strict preferences only and no ties (for illustrative purposes only);

Figure 5 is a flow diagram illustrating the main steps in the method and computer program of the invention;

Figure 6 is a table showing the ambiguities from the six profiles (excluding those that are unambiguous) for an exemplar APS with three criteria and two categories;

Figure 7 is a table of the total and unique ambiguities for different APSs and degrees;

Figure 8 is a continuation of the table from Figure 7 of the total and unique ambiguities for different APSs and degrees;

Figure 9 is a flow diagram illustrating the main components of the 'efficient ambiguities generator' described in Step 1 of the method of the invention;

Figure 10 is a table of the sufficient (but not necessary) conditions for implicitly resolving the 3rd-degree ambiguities of an exemplar APS with three criteria and two categories;

Figure 11 is a flow diagram illustrating an overview of Steps 1 to 5 of the computer program of the invention;

Figure 12 is a flow diagram illustrating the preferred main components of the ambiguity generator of the computer program;

Figure 13 is a flow diagram illustrating the preferred main components involved in testing whether or not an ambiguity is resolved for the computer program;

Figure 14 is a flow diagram illustrating the preferred main components involved in the explicit resolution of ambiguities for the computer program; and

Figure 15 is a flow diagram illustrating the preferred main components of the undo module (whereby explicitly resolved ambiguities are revised) of the computer program.

DETAILED DESCRIPTION OF THE PREFERRED FORMS

The invention is primarily embodied in the methodology set out below both by itself and as implemented through computing resources such as the preferred resources set out in Figure 3, by way of example. The invention is also embodied in the software used to implement the methodology and in any system comprising a combination of hardware and software used to implement the methodology.

In its most preferred form the invention is implemented on a personal computer or workstation operating under the control of appropriate operating and application software.

Figure 3 shows the preferred system architecture of a personal computer, workstation, or server on which the invention could be implemented. The computer system 300 typically comprises a central processor 302, a main memory 304, for example RAM, and an input/output controller 306. The computer system 300 may also comprise peripherals such as a keyboard 308, a pointing device 310, for example a mouse, touchpad, or trackball, a display or screen device 312, a mass storage memory 314, for example a hard disk, floppy disk or optical disc and an output device 316 such as a printer. The system 300 could also include a network interface card or controller 318 and/or a modem 320. The individual components of the system 300 could communicate through a system bus 322.

The method of the present invention will now be described with reference to several examples, beginning with an APS with just three criteria: a, b and c; and two categories on each: 1 and 2; such that there are six criterion-category variables: $a1$, $a2$, $b1$, $b2$, $c1$ and $c2$.

For example, if this were an APS for selecting immigrants, criterion a might be 'educational qualifications', b 'wealth' and c 'language proficiency'. Category 1 might be generically defined as 'low' and 2 as 'high'. Real immigration APSs, however, typically have at least twice as many criteria and categories as this simple example.

By definition, the values of these variables monotonically increase with the categories within each criterion so that the following 'inherent inequalities' hold: $a2 > a1$, $b2 > b1$ and $c2 > c1$.

Corresponding to all possible combinations of the two categories on the abc criteria, eight profiles, each with a total score equation, are represented by this system (where the three-digit numbers are symbolic representations only):

$$
\begin{aligned}
222 &= a2 + b2 + c2 \\
221 &= a2 + b2 + c1 \\
212 &= a2 + b1 + c2 \\
122 &= a1 + b2 + c2 \\
112 &= a1 + b1 + c2 \\
121 &= a1 + b2 + c1 \\
211 &= a2 + b1 + c1 \\
111 &= a1 + b1 + c1
\end{aligned}
$$

The essence of calibrating an APS such as this one is deciding the point values of the six variables ($a1$, $a2$, $b1$, $b2$, $c1$ and $c2$) such that decision makers' preferred (or 'valid') overall ranking of the eight profiles (equations) is realised.

The internal logic of APSs — specifically, the inviolable laws of arithmetic — restricts the otherwise 8! = 40,320 rankings (permutations), given strict preferences only and no ties (for illustrative purposes only), to the 12 rankings represented via the decision tree in Figure 4. The decision tree highlights the inherent contingencies in the derivations of the profile's rankings.

Any of the 12 rankings shown in Figure 4 can be produced from the six variables, depending on the values chosen for them. Ranking #1, for example, is given by $a1 = 0$, $a2 = 4$, $b1 = 0$, $b2 = 2$, $c1 = 0$ and $c2 = 1$, with the total scores: 222 = 7, 221 = 6, 212 = 5,

211 = 4, 122 = 3, 121 = 2, 112 = 1 and 111 = 0. Alternatively, ranking #12, for example, is given by $a1 = 0$, $a2 = 1$, $b1 = 0$, $b2 = 2$, $c1 = 0$ and $c2 = 4$, with the total scores: 222 = 7, 122 = 6, 212 = 5, 112 = 4, 221 = 3, 121 = 2, 211 = 1 and 111 = 0.

In general, any particular ranking of profiles is determined by their *pairwise* rankings vis-à-vis each other. For an APS with x criteria and y categories on each, and y^x profiles, a maximum of $\frac{y^x(y^x-1)}{2}$ pairwise rankings is possible. Thus for the exemplar APS with $x = 3$ and $y = 2$, and eight profiles (i.e., 2^3), there is a maximum of 28 [i.e., 8(8 – 1)/2] pairwise rankings.

The method of the invention minimises the number of pairwise rankings that must be decided *explicitly* (via value judgements), such that, in this example, a minimum of two and a maximum of four, rather than 28, is required, with the remaining pairwise rankings *implicitly* resolved as corollaries of the explicit rankings. The method of the invention comprises, in broad terms, three steps, as explained in turn below. The three main steps of the invention are illustrated in Figure 5.

Step 1 of the method of the invention 510 involves identifying (or 'generating') the 'ambiguities' of the APS that is being calibrated.

Ambiguities are formed from the total score equations of profile pairs whose pairwise rankings are a priori *ambiguous*, given the inherent inequalities (as already explained, $a2 > a1$, $b2 > b1$ and $c2 > c1$ in the present example). These are profile pairs in which one profile has a *higher* categorical rating on at least one criterion and a *lower* rating on at least one other criterion than the other profile.

For example, the pairwise ranking of profiles 221 and 212 — hereinafter referred to as "221 *vs* 212" and corresponding to $a2 + b2 + c1$ *vs* $a2 + b1 + c2$ — is ambiguous given $a2 > a1$, $b2 > b1$ and $c2 > c1$, as noted above.

On the other hand, many profile pairs are unambiguously ranked. For example, 222 ($a2 + b2 + c2$) is always pairwise ranked first and 111 ($a1 + b1 + c1$) is always pairwise ranked second — e.g., $a2 + b2 + c2 > a1 + b2 + c2$ (122) and $a1 + b2 + c2$ (122) $> a1 + b1 + c1$, and so on. Other profile pairs are similarly unambiguously ranked, for example, $a1 + b2 + c2 > a1 + b1 + c2$ (122 > 112), and so on.

As described above, identifying and eliminating all possible unambiguous pairwise rankings serves to cull the rankings of the eight possible profiles from 8! permutations (40,320) to 48 (for illustrative purposes only, given strict preferences only and no ties) from which the 12 in Figure 4 are further culled via another aspect of the internal logic of APSs described below.

The method of the invention identifies and excludes profile pairs that are unambiguously ranked and then focuses exclusively on profile pairs that are ambiguously ranked.

Some ambiguously ranked profile pairs can be 'reduced' by cancelling variables that are common to both profile equations. Thus $a2 + b2 + c1$ *vs* $a2 + b1 + c2$ (221 *vs* 212, as above) can be reduced to $b2 + c1$ *vs* $b1 + c2$ by cancelling $a2$ from both profiles' equations. In effect, because $a2$ appears in both equations it has no bearing on the ranking of the two profiles that the equations represent. We refer to such reduced forms as 'ambiguities'.

Moreover $b2 + c1$ *vs* $b1 + c2$ also corresponds to 121 *vs* 112, as $a1 + b2 + c1$ *vs* $a1 + b1 + c2$ reduces to $b2 + c1$ *vs* $b1 + c2$ after cancelling the $a1$ terms from both profiles' equations. Thus $b2 + c1$ *vs* $b1 + c2$ represents *two* ambiguously ranked profile pairs: 121 *vs* 112 *and* 221 *vs* 212.

However, not all ambiguously ranked profile pairs are reducible in this fashion. For example, no variables can be cancelled from $a2 + b2 + c1$ *vs* $a1 + b1 + c2$ (221 *vs* 112), as none are common to both profiles' equations. Nonetheless, such irreducible ambiguously ranked profile pairs are also hereinafter referred to as ambiguities.

Accordingly ambiguities can be classified by the number of criteria they contain, hereinafter referred to as the 'degree' of the ambiguity. Thus '2^{nd}-degree' ambiguities contain two criteria, for example, $b2 + c1$ *vs* $b1 + c2$, as above, and '3^{rd}-degree' ambiguities contain three criteria, for example, $a2 + b2 + c1$ *vs* $a1 + b1 + c2$, as above, and so on.

In general, the ambiguities for an APS with x criteria range from 2^{nd}-degree to x^{th}-degree.

The algorithmically simplest process for generating ambiguities is to first create all y^x combinations of the y categories on the x criteria (that is, all profiles), and then pairwise rank them all against each other to identify pairs in which one profile has a higher categorical rating on at least one criterion and a lower rating on at least one other criterion. There will always be a total of $\frac{y^x(y^x-1)}{2}$ pairwise rankings.

As each ambiguously ranked profile pair is identified, it is reduced where possible by canceling variables in both profiles' equations, and retained only if the resulting ambiguity has not already been discovered. In other words, replicated ambiguities are discarded.

Accordingly the ambiguities for the exemplar APS with $x = 3$ and $y = 2$ are set out in Figure 6, where the shaded ambiguities are duplicates of ones reported earlier in the table. In Figure 6, rankings that are unambiguous are denoted by 'n.a.'. The blank elements of the matrix, except for the main diagonal, are mirror images of the ones reported, and shaded ambiguities are duplicates of ones reported earlier.

Thus it can be seen in Figure 6 that although there are nine ambiguities in total, only six are unique:

(1) $b2 + c1$ *vs* $b1 + c2$
(2) $a2 + c1$ *vs* $a1 + c2$
(3) $a2 + b1$ *vs* $a1 + b2$

(4) $a2 + b2 + c1$ *vs* $a1 + b1 + c2$
(5) $a2 + b1 + c2$ *vs* $a1 + b2 + c1$
(6) $a1 + b2 + c2$ *vs* $a2 + b1 + c1$

Clearly ambiguities (1) to (3) are 2nd-degree ambiguities (and are duplicated in the figure) and ambiguities (4) to (6) are 3rd-degree ambiguities.

In general, the number of ambiguities of each degree can be calculated from the two equations described below.

For a given APS with x criteria and y categories on each criterion, the *total* number of ambiguities (including replicates) of a given degree (z) is given by:

$$\frac{x!}{(x-z)!z!} \times (2^{z-1} - 1) \times \left(\frac{y(y-1)}{2}\right)^{z} \times y^{x-z} \qquad (1)$$

The first term corresponds to ${}^{x}C_{z}$ (the combinations formula); the second term is the powers of 2 and minus one; the third term is ${}^{y}C_{2}$ raised to the z^{th}-power or, alternatively, the sum of the first $y - 1$ natural numbers (e.g., $5(5-1)/2 = 10 = 1 + 2 + 3 + 4$) raised to the z^{th}-power; and the last term is a power function.

Of these ambiguities, the number of *unique* ambiguities (excluding replicates) of a given degree is given by:

$$\frac{x!}{(x-z)!z!} \times (2^{z-1} - 1) \times \left(\frac{y(y-1)}{2}\right)^{z} \qquad (2)$$

The two equations (1) and (2) differ only by the term $(\times)\ y^{x-z}$. The meaning of this and the other three terms (common to both equations) is discussed further below when an alternative process for generating ambiguities is described.

Equations (1) and (2) can be compared with the equation mentioned earlier for the total number of pairwise rankings (including unambiguous ones): $\dfrac{y^{x}(y^{x}-1)}{2}$.

Equation (2) is particularly useful because it reveals how many ambiguities at each degree must be resolved for any given APS. Figures 7 and 8 report the numbers of total and unique ambiguities for different APSs and degrees. As illustrated there, the number of ambiguities can be relatively large even for small values of x and y. For example, an APS with $x = 6$ and $y = 4$ (twice the values for the exemplar APS above) has a total of 2,295,756 unique ambiguities across its five degrees (2^{nd} to 6^{th}).

The simple process described above for generating ambiguities is further simplified if any profiles that are theoretically impossible are culled from the set to be pairwise ranked before any ambiguities are generated.

For example, in the APS for hip or knee replacements referred to in Figures 1 and 2 it would be a contradiction for a patient to be rated as having "*severe*" "*Pain on motion (e.g., walking, bending)*" while also being rated on another of the criteria as having the "*Ability to walk without significant pain*" for a distance of "over 5 blocks". Such a combination of categories on these two criteria is theoretically impossible and therefore all profiles that include it could be deleted from the list to be pairwise ranked.

Similarly, when validating an extant APS (rather than calibrating a new one), the profiles to be evaluated can be determined from a 'stocktake' of the alternatives ranked by the APS over its lifetime. Any other profiles that might realistically be expected in the future could be added.

This process also serves to increase the efficiency of Steps 2 and 3 of the method of the invention described below, as in general the more profiles that are culled, the fewer ambiguities there are, and therefore the simpler is the calibration exercise.

Notwithstanding such refinements, the simple process described above is computationally inefficient because profile comparisons that are ultimately unnecessary are performed and replicated ambiguities are discarded.

For example, to generate the above-mentioned 2,295,756 ambiguities for an APS with $x = 6$ and $y = 4$, as can be calculated from the data in Figure 8, 6,090,804 unnecessary profile comparisons are performed and 5,094,900 ambiguities are discarded.

The particularly preferred process for generating ambiguities according to the invention is therefore described below. This is also the preferred process for the computer program of the invention. The main components of the process are illustrated in Figure 9.

The particularly preferred process, hereinafter referred to as the 'efficient ambiguities generator', is described with explicit reference to the three terms in equation (2) above (reproduced below) and a specific example. The example is that of generating the 3rd-degree ambiguities for an APS with $x = 5$ and $y = 3$, of which there are 810 in total (as can be calculated from equation (2) and is reported in Figure 7).

Although the equation applies to APSs with the same number of categories on the criteria, as explained later below, the process can be generalised to allow the number of categories to vary across criteria. For example, instead of $y = 3$ for all 5 criteria in the above-mentioned APS, as is applied later, criterion *a* could have two categories, *b* three categories, *c* four, and so on.

$$\frac{x!}{(x-z)!z!} \times (2^{z-1}-1) \times \left(\frac{y(y-1)}{2}\right)^z \qquad (2)$$

Equation (2)'s *first term* — $\frac{x!}{(x-z)!z!}$ ($= {}^xC_z$, the combinations formula) — is the number of combinations of z criteria that can be selected from x criteria.

For the above-mentioned APS with $x = 5$ (that is, criteria *a*, *b*, *c*, *d* and *e*), 10 combinations of three criteria ($z = 3$) are possible: *abc, abd, abe, acd, ace, ade, bcd, bce, bde* and *cde*. Well-known algorithms exist for generating such combinations.

Each combination can be thought of as forming the 'base' for a group of ambiguities. For example, *abc* is the base for all 3rd-degree ambiguities centred on $a + b + c$ *vs* $a + b + c$ (hereinafter abbreviated to *abc vs abc*), such as $a2 + b3 + c3$ *vs* $a3 + b2 + c2$.

Because the criteria in each base (10 of them in the present example) have the same number of categories each, the bases can be treated identically with respect to the following operations that correspond to the other two terms in equation (2).

The *second term* $(2^{z-1} - 1)$ can be interpreted as representing the number of underlying 'structures' for a given degree (z). Structures are generated by first listing the numbers between 1 and $2^{z-1} - 1$ in binary form using z bits, where each bit represents a criterion of two categories represented by either 0 or 1.

Continuing with the example of generating the 3rd-degree ambiguities for an APS with $x = 5$, the structures for $z = 3$ are:

011
010
001

These structures correspond to 3, 2, and 1 in binary form. For now, each structure can be thought of as being analogous to a profile with $y = 2$ for all criteria. Therefore only one ambiguity can be created from each, which is done by finding each structure's '1s complement', that is, by 'flipping the bits'; thus in the example:

011 *vs* 100
010 *vs* 101
001 *vs* 110

Each term (either a '0' or a '1') on either side of the ambiguity structure represents a criterion, and for each criterion their relative magnitudes ('0' versus '1') represents the relative magnitude of the categories they represent (i.e., 'low' or 'high'), such that each structure has an underlying pattern.

Thus 011 *vs* 100 in the example above (and analogously for 010 *vs* 101 and 001 *vs* 110) signifies that the first of the three criteria represented (corresponding to 0_ _ *vs* 1_ _) has a *lower* category on the left hand side (LHS) and a *higher* category on the right hand side (RHS) of each ambiguity that is derived from it.

Similarly, the second and third criteria (corresponding to _11 *vs* _00) both have *higher* categories on the LHS and *lower* categories on the RHS.

The equation's *third term* — $\left(\frac{y(y-1)}{2}\right)^z$ — is the number of ambiguities that can be generated from each ambiguity structure, as determined by the number of categories (y) on the criteria in a given base. There are three steps to generating these ambiguities, as follows.

First, all of the bases are matched with all of the structures. In the present example, the 10 bases (*abc*, *abd*, *abe*, *acd*, *ace*, *ade*, *bcd*, *bce*, *bde* and *cde*) are matched with the three structures (011 *vs* 100, 010 *vs* 101 and 001 *vs* 110) to produce $10 \times 3 = 30$ matches:

abc vs abc with 011 *vs* 100
abc vs abc with 010 *vs* 101
abc vs abc with 001 *vs* 110
abd vs abd with 011 *vs* 100
... and so on for another 26 matches.

Second, the underlying 'pattern' for each match (that is, a base with a structure) is implemented, according to the number of categories on the bases.

For *abc vs abc* with 011 *vs* 100, for example, with three categories on the criteria ($y = 3$), there are *three* ways each (that is, $\frac{y(y-1)}{2} = 3(3-1)/2$ ways) of representing criterion *a* with a *lower* category on the LHS and a *higher* category on the RHS (0_ _ *vs* 1_ _), and criteria *b* and *c* both with *higher* categories on the LHS and *lower* categories on the RHS (_11 *vs* _00):

*a*2 *vs a*3 *b*3 *vs b*2 *c*3 *vs c*2

$a1$ *vs* $a3$	$b3$ *vs* $b1$	$c3$ *vs* $c1$
$a1$ *vs* $a2$	$b2$ *vs* $b1$	$c2$ *vs* $c1$

Note that $\frac{y(y-1)}{2} = {}^{y}C_2$, the number of combinations of categories (within the criterion) taken two-at-a-time. The output listed above embodies these combinations, where the two categories are simply ordered as required by the structure.

Finally, all $3 \times 3 \times 3 = 27$ possible combinations — i.e., $\left(\frac{y(y-1)}{2}\right)^z = [3(3-1)/2]^3$ — of these three patterns are formed, thereby obtaining all 27 3rd-degree ambiguities corresponding to the match *abc vs abc* with 011 *vs* 100:

$a2 + b3 + c3$ *vs* $a3 + b2 + c2$
$a2 + b3 + c3$ *vs* $a3 + b2 + c1$
$a2 + b3 + c2$ *vs* $a3 + b2 + c1$
$a2 + b3 + c3$ *vs* $a3 + b1 + c2$
$a2 + b3 + c3$ *vs* $a3 + b1 + c1$
$a2 + b3 + c2$ *vs* $a3 + b1 + c1$
$a2 + b2 + c3$ *vs* $a3 + b1 + c2$
$a2 + b2 + c3$ *vs* $a3 + b1 + c1$
$a2 + b2 + c2$ *vs* $a3 + b1 + c1$
$a1 + b3 + c3$ *vs* $a3 + b2 + c2$
$a1 + b3 + c3$ *vs* $a3 + b2 + c1$
$a1 + b3 + c2$ *vs* $a3 + b2 + c1$
$a1 + b3 + c3$ *vs* $a3 + b1 + c2$
$a1 + b3 + c3$ *vs* $a3 + b1 + c1$
$a1 + b3 + c2$ *vs* $a3 + b1 + c1$
$a1 + b2 + c3$ *vs* $a3 + b1 + c2$
$a1 + b2 + c3$ *vs* $a3 + b1 + c1$
$a1 + b2 + c2$ *vs* $a3 + b1 + c1$
$a1 + b3 + c3$ *vs* $a2 + b2 + c2$
$a1 + b3 + c3$ *vs* $a2 + b2 + c1$
$a1 + b3 + c2$ *vs* $a2 + b2 + c1$
$a1 + b3 + c3$ *vs* $a2 + b1 + c2$
$a1 + b3 + c3$ *vs* $a2 + b1 + c1$
$a1 + b3 + c2$ *vs* $a2 + b1 + c1$
$a1 + b2 + c3$ *vs* $a2 + b1 + c2$
$a1 + b2 + c3$ *vs* $a2 + b1 + c1$
$a1 + b2 + c2$ *vs* $a2 + b1 + c1$

Analogues of the three steps explained above are performed for all matches. For each of the 30 matches in the present example with $x = 5$, $y = 3$ and $z = 3$, 27 ambiguities analogous to the 27 above are generated, resulting in a total of $30 \times 27 = 810$ (unique) 3rd-degree ambiguities — i.e., $\frac{x!}{(x-z)!z!} \times (2^{z-1} - 1) \times \left(\frac{y(y-1)}{2}\right)^z = 10 \times 3 \times 27 = 810$.

As noted earlier, the process outlined above can be generalised to allow the number of categories to vary across the criteria. The key difference from the process explained above is that the underlying 'pattern' for each match (for example, *abc vs abc* with 011 *vs* 100) is idiosyncratic to the criteria included, as determined by the numbers of categories on each criterion.

For example, if in the example referred to above, instead of three categories on all criteria, criterion *a* has two categories, *b* has three and *c* four. Thus with just two categories for criterion *a* (for 011 *vs* 100, corresponding to 0_ _ *vs* 1_ _), there is only one underlying pattern ($\frac{y(y-1)}{2}$ = 2(2 – 1)/2) corresponding to a *lower* category on the LHS and a *higher* category on the RHS of each ambiguity that is derived:

*a*1 *vs a*2

As before, with three categories for criterion *b* (corresponding to _1_ *vs* _0_), there are three underlying patterns ($\frac{y(y-1)}{2}$ = 3(3 – 1)/2) corresponding to a *higher* category on the LHS and a *lower* category on the RHS:

*b*3 *vs b*2
*b*3 *vs b*1
*b*2 *vs b*1

Finally, with four categories for criterion *c* (corresponding to _ _1 *vs* _ _0), there are six underlying patterns ($\frac{y(y-1)}{2}$ = 4(4 – 1)/2) corresponding to a *higher* category on the LHS and a *lower* category on the RHS:

*c*4 *vs c*3

$c4$ *vs* $c2$
$c4$ *vs* $c1$
$c3$ *vs* $c2$
$c3$ *vs* $c1$
$c2$ *vs* $c1$

By taking all $1 \times 3 \times 6 = 18$ combinations of the above three sets of criteria-categories, all 18 3rd-degree ambiguities corresponding to *abc vs abc* with 011 *vs* 100 (with two, three and four categories respectively) may be obtained:

$$a1 + b3 + c4 \text{ vs } a2 + b2 + c3$$
$$a1 + b3 + c4 \text{ vs } a2 + b2 + c2$$
$$a1 + b3 + c4 \text{ vs } a2 + b2 + c1$$
$$a1 + b3 + c3 \text{ vs } a2 + b2 + c2$$
$$a1 + b3 + c3 \text{ vs } a2 + b2 + c1$$
$$a1 + b3 + c2 \text{ vs } a2 + b2 + c1$$
$$a1 + b3 + c4 \text{ vs } a2 + b1 + c3$$
$$a1 + b3 + c4 \text{ vs } a2 + b1 + c2$$
$$a1 + b3 + c4 \text{ vs } a2 + b1 + c1$$
$$a1 + b3 + c3 \text{ vs } a2 + b1 + c2$$
$$a1 + b3 + c3 \text{ vs } a2 + b1 + c1$$
$$a1 + b3 + c2 \text{ vs } a2 + b1 + c1$$
$$a1 + b2 + c4 \text{ vs } a2 + b1 + c3$$
$$a1 + b2 + c4 \text{ vs } a2 + b1 + c2$$
$$a1 + b2 + c4 \text{ vs } a2 + b1 + c1$$
$$a1 + b2 + c3 \text{ vs } a2 + b1 + c2$$
$$a1 + b2 + c3 \text{ vs } a2 + b1 + c1$$
$$a1 + b2 + c2 \text{ vs } a2 + b1 + c1$$

This process is performed for all matches. But, because the numbers of categories for the criteria are different, each match generates an idiosyncratic set of ambiguities that depends on the numbers of categories on the included criteria.

For example, the number of ambiguities generated from *abc vs abc* with 011 *vs* 100 (as above) is different to the number from *abd vs abd* with 011 *vs* 100 when the numbers of categories on criteria c and d are different. Therefore, in general the ambiguities for each of the $\frac{x!}{(x-z)!z!}$ bases (combinations of z criteria from the x criteria) must be generated individually.

For example, in the case of the base corresponding to the particular combination of z criteria comprising *the first* z of the x criteria (*criterion 1*, *criterion 2*, ... *criterion z*, where $z \leq x$), there are $2^{z-1}-1$ structures as before. (The choice of *the first* z of the x criteria, instead of any other z of the x criteria, is for notational simplicity.) Each structure has $\left(\frac{y_1(y_1-1)}{2}\right)\times\left(\frac{y_2(y_2-1)}{2}\right)\times\ldots\times\left(\frac{y_z(y_z-1)}{2}\right)$ ambiguities, where $y_1, y_2 \ldots y_z$ are the numbers of categories on each of these first z criteria.

The total number of z^{th}-degree ambiguities is obtained by summing the number of ambiguities (analogous to the example) across all $\frac{x!}{(x-z)!z!}$ bases. In general, the number of ambiguities of each degree can be calculated from the equation described below.

For a given APS with $y_1, y_2, \ldots y_x$ categories on the respective x criteria, the equation is based on the following definitions. First, Y is the set of the numbers of categories on the x criteria: $Y = \{y_1, y_2, \ldots y_x\}$. Second, C is the set of unordered z-tuples formed by taking all possible $\frac{x!}{(x-z)!z!}$ combinations of the elements of Y, z-at-a-time: $C = \{c \mid c$ is an unordered z-tuple from Y, as defined above$\}$. Each of C's elements (i.e., sets), c_i, is numbered from 1 to $\frac{x!}{(x-z)!z!}$: c_i, $i = 1, 2, \ldots \frac{x!}{(x-z)!z!}$. Each of c_i's elements, y_{ij}, is numbered from 1 to z: y_{ij}, $j = 1, 2, \ldots z$.

Applying these definitions, the number of unique ambiguities of a given degree (z), when the number of categories varies across criteria, is given by:

$$(2^{z-1}-1)\times\sum_{i=1}^{\frac{x!}{(x-z)!z!}}\prod_{j=1}^{z}\frac{y_{ij}(y_{ij}-1)}{2} \qquad (3)$$

Alternatively this equation can be expressed as: $(2^{x-1}-1)\times\sum_{c\in C}\prod_{y\in c}\frac{y(y-1)}{2}$.

For common values of y_{ij} ($y_{ij}=y$), that is, all criteria have the same numbers of categories, equation (3) is equivalent to equation (2) above.

Either process explained above — for the same or, alternatively, different numbers of categories on the criteria — can be used to generate all of an APS's ambiguities of a given degree (such as 3rd-degree, as above), or, alternatively, ambiguities can be generated individually.

As each ambiguity is generated it is tested for whether or not it is theoretically impossible, and therefore to be discarded or not, given the theoretically impossible profiles that were culled earlier (as described above). For the simple process for generating ambiguities described earlier, all theoretically impossible profiles are simply removed before ambiguities are generated from them.

However, because the 'efficient ambiguities generator' explained immediately above does not generate ambiguities from profiles, the ambiguities must be tested as they are generated for whether or not the profile pairs that they represent have been culled or not.

For example, for an APS with $x = 4$ and $y = 3$, the reduced form $a1 + b3$ *vs* $a3 + b2$ (i.e., 13_ _ *vs* 32_ _) represents nine ambiguously ranked profile pairs:

1311 *vs* 3211
1312 *vs* 3212
1313 *vs* 3213
1321 *vs* 3221
1322 *vs* 3222
1323 *vs* 3223
1331 *vs* 3231
1332 *vs* 3232
1333 *vs* 3233

For $a1 + b3$ *vs* $a3 + b2$ to be theoretically impossible and therefore discardable, in all nine of these pairs at least one of the profiles — a minimum of nine and a maximum of 18 profiles — must be theoretically impossible. If instead at least one of the nine pairs is not excluded, then $a1 + b3$ *vs* $a3 + b2$ is possible and therefore ought not to be discarded. Accordingly all nine pairs must be considered before it can be determined whether $a1 + b3$ *vs* $a3 + b2$ should be discarded or not, but as soon as one profile pair is found that has not been excluded, the ambiguity should be retained.

Enumerating all such profile pairs for any given ambiguity of degree z is relatively straight-forward, as each profile pair is based on the ambiguity in question, augmented by all possible combinations of the categories on the other $x - z$ criteria. There are therefore y^{x-z} such profile pairs when the number of categories on the criteria (y) is the same.

As noted earlier, the term y^{x-z} appears in equation (1) above — giving the *total* number of ambiguities (including replicates) of a given degree — but not in the otherwise identical equation (2) — giving the number of *unique* ambiguities (excluding replicates) of a given degree.

Accordingly y^{x-z} can be interpreted as the number of 'copies' of a particular ambiguity generated by the algorithmically simple process explained earlier. That is, in the example above with $x = 4$ and $y = 3$, each of the $y^{x-z} = 3^{4-2} = 9$ pairwise profile comparisons generates $a1 + b3$ *vs* $a3 + b2$ (of which eight are discarded because they are replicates).

Having generated the ambiguities, Step 2 of our method involves *explicitly* resolving them, one-at-a-time, while identifying all other ambiguities that are *implicitly* resolved as corollaries. This step is represented in Figure 5 at 520.

Any of the unresolved ambiguities of any degree could be selected for explicit resolution during the calibration process, however there tend to be fewer ambiguities to store at lower degrees, and lower degree ambiguities are easier for decision makers to resolve, and so the process starts by resolving the 2nd-degree ambiguities and then proceeds to

resolving successively higher-degree ambiguities. If instead it were desired that the number of decisions be minimised at the expense of the complexity of the decisions, the process should instead start by resolving the highest degree ambiguities and then proceed to resolving successively lower-degree ambiguities.

As ambiguities can only be resolved via value judgements, they must be decided by an individual 'decision maker' or group of 'decision makers', preferably with knowledge of the field in which the particular APS is to be applied. For example, a group of decision makers for a medical APS may comprise a panel of doctors and patients. Hereinafter decision makers (plural) are referred to.

Accordingly the preferences of the decision makers must be probed via a series of questions concerning their pairwise rankings of profiles.

As listed earlier, the ambiguities for the original exemplar APS with $x = 3$ and $y = 2$ are: (1) $b2 + c1$ *vs* $b1 + c2$, (2) $a2 + c1$ *vs* $a1 + c2$, (3) $a2 + b1$ *vs* $a1 + b2$, (4) $a2 + b2 + c1$ *vs* $a1 + b1 + c2$, (5) $a2 + b1 + c2$ *vs* $a1 + b2 + c1$ and (6) $a1 + b2 + c2$ *vs* $a2 + b1 + c1$.

For ambiguity (1) $b2 + c1$ *vs* $b1 + c2$, for example, the decision makers are asked, in essence: "Given two alternatives that are the same with respect to criterion a, which has the greater priority, _21 or _12?"

If there is more than one decision maker, the process of getting answers to this and subsequent questions can be streamlined by asking decision makers to cast votes (perhaps via email) for the pairwise rankings they favour.

However, in this type of situation the majority voting runs the risks of the well-known voting paradox, whereby, depending on the decision makers' *individual* rankings of the profiles, the order the ambiguities are voted on can determine the resolutions that are derived. To avoid this possibility, were it likely, the decision makers should instead be required to reach a consensus on their pairwise rankings.

Logically, three mutually exclusive and exhaustive answers to the above question are possible: (1) _21 is strictly preferred to _12 or (2) _12 is strictly preferred to _21 or (3) they are equally preferred (i.e., indifference between _21 and _12).

Notationally, these three preferences can be represented as (1) _21 > _12 or (2) _12 > _21 or (3) _21 = _12, corresponding to (1) $b2 + c1 > b1 + c2$ or (2) $b1 + c2 > b2 + c1$ or (3) $b2 + c1 = b1 + c2$ (where, as usual, ">" is "strictly greater than" and "=" is "equal to").

Weak preferences, for example, _21 is at least as preferred as _12 (notationally, _21 ≥ _12), are also a logical possibility. However strict preferences are more useful from a practical perspective.

The decision makers might, quite naturally, protest that their answer to the above question ("Given two alternatives that are the same with respect to criterion *a*, which has the greater priority, _21 or _12?") depends on whether criterion *a* is rated '1' or '2' (for both alternatives). Nonetheless, such distinctions are precluded by the internal logic of APSs. If the decision makers will not answer the question as it is posed then, in effect, the APS itself will answer it by default, since the ambiguity will eventually be implicitly resolved by the other explicitly resolved ambiguities chosen by the decision makers.

More specifically, for example, *if* the decision makers were to decide that 121 > 112 — corresponding to $b2 + c1 > b1 + c2$ — then this implies 221 > 212, and vice versa. And analogously for $b1 + c2 > b2 + c1$ and $b1 + c2 = b2 + c1$.

The key word above is "*if*", as clearly neither inequality holds intrinsically. Therefore a value judgement is required to resolve ambiguity (1) $b2 + c1$ *vs* $b1 + c2$: either 121 > 112 *and* 221 > 212 or 112 > 121 *and* 212 > 221 or 121 = 112 *and* 221 = 212. By virtue of the laws of arithmetic, if one inequality or equality holds then the other must too; if one does not then neither does the other. It is impossible to have one half of either proposition without the other.

Accordingly the question above could be rephrased, in essence, as: "Which one of the following three possible rankings of two alternatives do you prefer, (1) 121 > 112 *and* 221 > 212 or (2) 112 > 121 *and* 212 > 221 or (3) 121 = 112 *and* 221 = 212?"

Continuing with the example, suppose that in fact the decision makers resolve ambiguity (1) by choosing _21 > _12 (in other words 121 > 112 *and* 221 > 212), corresponding to $b2 + c1 > b1 + c2$. Two alternative, but equivalent, approaches are available for identifying the implicitly resolved ambiguities.

Although these approaches — hereinafter referred to as 'Approach 1' (of which there are two variants) and 'Approach 2' — differ in the means and the sequence in which the implicitly resolved ambiguities are identified, both generate the *same* list of explicitly resolved ambiguities, from which the point values are derived at Step 3 described below.

However because Approach 1 becomes relatively unwieldy, and therefore more resource intensive to implement, for more criteria and categories than the exemplar APS with $x = 3$ and $y = 2$, the particularly preferred embodiment of the computer program of the invention is based on Approach 2. Nonetheless, Approach 1 is described first as it is more intuitively tractable and is therefore useful for illustrating Approach 2.

Approach 1 may be summarised as follows. After a given ambiguity is explicitly resolved by the decision makers, *all* other ambiguities that are implicitly resolved as corollaries are immediately identified by adding appropriate inherent inequalities and/or other explicitly resolved ambiguities (inequalities or equalities). Then another (unresolved) ambiguity is explicitly resolved by the decision makers and all its corollaries are identified. The process is repeated until all ambiguities have been resolved, either explicitly or implicitly.

Thus a corollary of the *explicit* resolution of ambiguity (1) as $b2 + c1 > b1 + c2$ (decided by the decision makers, as explained earlier) is the *implicit* resolution of ambiguity (4) $a2$

$+ b2 + c1$ *vs* $a1 + b1 + c2$. This is revealed by *adding* the inherent inequality $a2 > a1$ and $b2 + c1 > b1 + c2$: $(a2 > a1) + (b2 + c1 > b1 + c2) = (a2 + b2 + c1 > a1 + b1 + c2)$.

Although this addition is mathematically legitimate, its theoretical validity in the context of APSs rests on the assumption that the decision makers are logically consistent in the sense that their pairwise profile rankings are transitive. Transitivity means in general that if alternatives $A > B$ and $B > C$, then $A > C$.

Thus, in the present example, the *explicit* resolution of ambiguity (1) as $b2 + c1 > b1 + c2$ corresponds to 121 > 112 and 221 > 212, as discussed earlier. Moreover 212 > 112 because $a2 > a1$. Therefore, assuming profile rankings are transitive, 221 > 212 and 212 > 112 implies 221 > 112 — corresponding to $a2 + b2 + c1 > a1 + b1 + c2$, as was revealed above by adding $a2 > a1$ and $b2 + c1 > b1 + c2$.

As there are no other corollaries at this point, the next ambiguity on the list, in this case ambiguity (2) $a2 + c1$ *vs* $a1 + c2$, can be presented to the decision makers to resolve via an analogous question to the first one: "Given two alternatives that are the same with respect to criterion *b*, which has the greater priority, 1_2 or 2_1?"

Suppose the decision makers answer 1_2 > 2_1, corresponding to $a1 + c2 > a2 + c1$. A corollary is the implicit resolution of ambiguity (6) $a1 + b2 + c2$ *vs* $a2 + b1 + c1$, as revealed by adding inherent inequality $b2 > b1$ to $a1 + c2 > a2 + c1$: $(b2 > b1) + (a1 + c2 > a2 + c1) = (a1 + b2 + c2 > a2 + b1 + c1)$.

In addition, both (1) $b2 + c1 > b1 + c2$ and (2) $a1 + c2 > a2 + c1$ implicitly resolve ambiguity (3) $a2 + b1$ *vs* $a1 + b2$, as revealed by their addition: $(b2 + c1 > b1 + c2) + (a1 + c2 > a2 + c1) = (a1 + b2 > a2 + b1)$ (after cancelling the *c* terms).

As for the addition of inherent inequalities and explicitly resolved ambiguities (as above), this addition is also justified by the assumption that the pairwise profile rankings of the decision makers are transitive.

Thus, in the present example, the *explicit* resolution of ambiguity (1) as $b2 + c1 > b1 + c2$ corresponds to 121 > 112 and 221 > 212, and (2) $a1 + c2 > a2 + c1$ corresponds to 112 > 211 and 122 > 221. Given 121 > 112 and 112 > 211, then, by transitivity, 121 > 211 is implied. Likewise, given 122 > 221 and 221 > 212, then 122 > 212 is implied. Both 121 > 211 and 122 > 212 correspond to $a1 + b2 > a2 + b1$, as was revealed above by adding (1) $b2 + c1 > b1 + c2$ and (2) $a1 + c2 > a2 + c1$.

Thus from just two explicit decisions to resolve ambiguities (1) and (2), another three ambiguities (3, 4 and 5) are implicitly resolved, so that five of the six ambiguities are resolved.

The remaining ambiguity, ambiguity (5) $a2 + b1 + c2$ *vs* $a1 + b2 + c1$, must be explicitly resolved by the decision makers, via a question that is conceptually simpler than the two earlier ones: "Which alternative has the greater priority, 212 or 121?" Suppose the decision makers answer 212 > 121, corresponding to $a2 + b1 + c2 > a1 + b2 + c1$.

The system is now fully specified as: (1) $b2 + c1 > b1 + c2$, (2) $a1 + c2 > a2 + c1$, (3) $a1 + b2 > a2 + b1$, (4) $a2 + b2 + c1 > a1 + b1 + c2$, (5) $a2 + b1 + c2 > a1 + b2 + c1$ and (6) $a1 + b2 + c2 > a2 + b1 + c1$ — as well as the inherent inequalities $a2 > a1$, $b2 > b1$ and $c2 > c1$. Of inequalities (1) to (6), only three (1, 2 and 5) were explicitly resolved by the decision makers, with the other three (3, 4 and 6) implicitly resolved as corollaries.

Finally, in general but not in the present example, any explicitly resolved ambiguities that are themselves corollaries of other explicitly resolved ambiguities can be removed from the list from which the point values are derived at Step 3 (below). This is because only independent inequalities/equalities are required for deriving point values.

A variant of the approach outlined above — in effect, its converse — is the identification of the *sufficient* (but not necessary) *conditions* for implicitly resolving ambiguities, in terms of (other) resolved ambiguities. Ambiguities are either explicitly resolved by the

decision makers or implicitly resolved when their sufficient conditions are met. Any ambiguities whose sufficient conditions are not met must therefore be resolved explicitly by the decision makers, until all ambiguities have been resolved, either explicitly or implicitly.

Thus in the exemplar APS with $x = 3$ and $y = 2$, ambiguities (4), (5) and (6) have four sufficient conditions each in terms of resolved 2nd-degree ambiguities. Specifically, ambiguity (4) is (implicitly) resolved as $a2 + b2 + c1 > a1 + b1 + c2$ if at least one of the following inequalities holds: $a2 + c1 > a1 + c2$ or $a2 + c1 = a1 + c2$ (in both cases because $b2 > b1$) or $b2 + c1 > b1 + c2$ or $b2 + c1 = b1 + c2$ (in both cases because $a2 > a1$).

These and analogous sufficient (but not necessary) conditions for ambiguities (5) and (6) to be implicitly resolved as $a2 + b1 + c2 > a1 + b2 + c1$ and $a1 + b2 + c2 > a2 + b1 + c1$ respectively are listed in Figure 10. No sufficient conditions exist for the opposite resolutions of the three ambiguities nor for equalities (that is, not for RHS > LHS nor LHS = RHS) in terms of resolved 2nd-degree ambiguities.

It is then simply a matter of comparing these conditions against resolved ambiguities (1), (2) and (3) (arrived at via Approach 1 explained above). Accordingly, as identified via shading in Figure 10, (1) $b2 + c1 > b1 + c2$ implicitly resolves ambiguity (4) as $a2 + b2 + c1 > a1 + b1 + c2$, and both (2) $a1 + c2 > a2 + c1$ and (3) $a1 + b2 > a2 + b1$ implicitly resolve ambiguity (6) as $a1 + b2 + c2 > a2 + b1 + c1$.

These are the same 3rd-degree resolutions as were revealed earlier via the explicit resolutions of ambiguities (1) and (2) and their additions to inherent inequalities and each other respectively. As such they reflect, as before, pairwise profile rankings that are transitive.

Finally, as before, ambiguity (5) remains to be explicitly resolved, and then the system is fully specified.

This variant of Approach 1 generalises for APSs with higher-degree ambiguities. Their sufficient conditions are in terms of both individual lower-degree resolved ambiguities and combinations of them, of which there may be many. For example, the sufficient conditions for the resolution of a 5^{th}-degree ambiguity are in terms of combinations of 2^{nd}-degree resolved ambiguities, 3^{rd}-degree resolved ambiguities and 4^{th}-degree resolved ambiguities.

However, sufficient conditions for an ambiguity of a given degree can also be identified in terms of resolved ambiguities of the same, or even higher, degrees. In the present example, sufficient conditions can also be identified for resolving 2^{nd}-degree ambiguities in terms of other resolved 2^{nd}-degree ambiguities. For example, a sufficient condition for resolving ambiguity (3) $a2 + b1$ *vs* $a1 + b2$ as $a1 + b2 > a2 + b1$ is (1) $b2 + c1 > b1 + c2$ *and* (2) $a1 + c2 > a2 + c1$, corresponding to $(b2 + c1 > b1 + c2) + (a1 + c2 > a2 + c1)$, as explained earlier.

This means that in general it is difficult to enumerate and check *all* possible sufficient conditions (to ensure that none are missed); therefore in practice, this variant must be supplemented by other methods, such as the first variant of Approach 1 explained above.

Finally, with respect to both variants of Approach 1, because ambiguities (1) to (3) are not independent, both the order and the manner in which they are resolved affects the number of explicit value judgements that are required. The maximum number required is four and the minimum is two.

For example, if inequality (3) had been decided *before* inequalities (1) and (2) (instead of after, as above), then all three ambiguities (as well as ambiguity 5) would have had to have been resolved explicitly. This is because inequality (3) $a1 + b2 > a2 + b1$ cannot be added to (1) $b2 + c1 > b1 + c2$ or (2) $a1 + c2 > a2 + c1$ to obtain the other inequality, and yet inequalities (1) and (2) imply (3). For ambiguities (4) to (6), on the other hand, one and only one must be resolved explicitly.

Alternatively, for example, had ambiguity (1) been resolved as $b2 + c1 = b1 + c2$ instead of $b2 + c1 > b1 + c2$ (indifference rather than strict preference), and ambiguity (2) resolved as $a1 + c2 > a2 + c1$ (as above), then no other explicit resolutions would have been necessary.

This can be confirmed by noting that (1) $b2 + c1 = b1 + c2$ implies both (4) $a2 + b2 + c1 > a1 + b1 + c2$ and (5) $a2 + b1 + c2 > a1 + b2 + c1$. In other words $(b2 + c1 = b1 + c2) + (b2 > b1)$ for both of them and (2) $a1 + c2 > a2 + c1$ implies (6) $a1 + b2 + c2 > a2 + b1 + c1$ (as above) and $(b2 + c1 = b1 + c2) + (a1 + c2 > a2 + c1)$ implies (3) $a1 + b2 > a2 + b1$. Clearly, the point values derived from this system of equations and inequalities would be different to the point values derived earlier.

Similarly the system would be completely specified by the explicit resolution of ambiguities (1) and (6) as $b1 + c2 > b2 + c1$ and $a2 + b1 + c1 > a1 + b2 + c2$ only. Of particular interest is: $(a2 + b1 + c1 > a1 + b2 + c2) + (b2 > b1) =$ (2) $(a2 + c1 > a1 + c2)$; and $(a2 + b1 + c1 > a1 + b2 + c2) + (c2 > c1) =$ (3) $(a2 + b1 > a1 + b2)$. This illustrates the fact that it is not necessary to resolve lower-degree (here 2nd-degree) ambiguities *before* implicitly resolving higher- degree (here 3rd-degree) ambiguities. The process can be reversed, as illustrated here.

Unfortunately, as mentioned earlier, both variants of Approach 1 become unwieldy for APSs with more criteria and categories than the exemplar APS with $x = 3$ and $y = 2$. This is because calculating and managing *all* possible combinations of additions or possible sufficient conditions is resource intensive.

The 'additions' variant of Approach 1 involves maintaining a list of all possible inequalities/equalities that result from the addition of each explicitly resolved ambiguity with each other explicitly resolved ambiguity as well as with the inherent inequalities, and with every sum generated, and each of these sums with each other recursively until all possible additive combinations are exhausted. Each newly generated ambiguity is

'checked off' against the list of implicitly resolved ambiguities: if it is not on the list then it is yet to be resolved.

Similarly, the 'sufficient conditions' variant of Approach 1 involves managing a list of all implicitly resolved ambiguities, identifying all possible sufficient conditions of lower degrees, of which there may be a great many combinations, and identifying ambiguities implied by the same or higher degree explicitly resolved ambiguities (by some other means).

Approach 2, on the other hand, which is explained below, is more efficient and is therefore the preferred method used in the computer program of the invention.

In summary, Approach 2 involves testing ambiguities individually for whether or not they are implicitly resolved as corollaries of the explicitly resolved ambiguities (that were resolved earlier). If a given ambiguity is identified as having been implicitly resolved then it is deleted. If instead it is not implicitly resolved then it must be explicitly resolved by the decision makers. The process is repeated until all ambiguities have been identified as having been implicitly resolved or they are explicitly resolved.

Thus, with reference to the exemplar APS with $x = 3$ and $y = 2$, after the decision makers resolve ambiguity (1) as $b2 + c1 > b1 + c2$ as described earlier, the next ambiguity on the list ((2) $a1 + c2$ *vs* $a2 + c1$) is tested for whether or not it is implicitly resolved as a corollary of $b2 + c1 > b1 + c2$, as well as the inherent inequalities $a2 > a1$, $b2 > b1$ and $c2 > c1$.

Here, as in the computer program of the invention, this and subsequent tests can be performed via linear programming. In effect, this test is performed by asking the following two hypothetical questions (of the method, not the decision makers).

Hypothetical Question 1: *If* it were the case that the ambiguity in question [here (2) $a1 + c2$ *vs* $a2 + c1$] had been implicitly resolved as LHS > RHS (i.e., $a1 + c2 > a2 + c1$), then

does a solution exist to the system comprising this hypothetical inequality and the (actual) explicitly resolved inequalities/equalities [here (1) $b2 + c1 > b1 + c2$] — as well as the inherent inequalities [here $a2 > a1$, $b2 > b1$ and $c2 > c1$]? (*Yes* or *No*?)

If the answer is *No* — and therefore it would not be theoretically possible for the decision makers, if they wanted to, to explicitly resolve the ambiguity in question as LHS > RHS (here $a1 + c2 > a2 + c1$) — then it must be true that either RHS > LHS or LHS = RHS (i.e., either $a2 + c1 > a1 + c2$ or $a1 + c2 = a2 + c1$). This implies that the ambiguity in question (here ambiguity 2) has been implicitly resolved (i.e., as either $a2 + c1 > a1 + c2$ or $a1 + c2 = a2 + c1$). Hence it is of no further use and can be deleted.

If instead the answer to Question 1 is *Yes* — and therefore it would be theoretically possible for the decision makers, if they wanted to, to explicitly decide LHS > RHS (here $a1 + c2 > a2 + c1$) — then the following second hypothetical question is asked.

Hypothetical Question 2: *If* it were instead the case that the ambiguity in question (here ambiguity 2) had been implicitly resolved as RHS > LHS (that is, $a2 + c1 > a1 + c2$), then does a solution exist to the system comprising this hypothetical inequality and the (actual) explicitly resolved inequalities/equalities [here (1) $b2 + c1 > b1 + c2$, as before] — as well as the inherent inequalities [here $a2 > a1$, $b2 > b1$ and $c2 > c1$]? (*Yes* or *No*?)

If the answer is *No* — and therefore it would not be theoretically possible for the decision makers, if they wanted to, to explicitly resolve the ambiguity in question as RHS > LHS (here $a2 + c1 > a1 + c2$) — then it must be true that either LHS > RHS or LHS = RHS (either $a1 + c2 > a2 + c1$ or $a1 + c2 = a2 + c1$). This implies that the ambiguity in question (here ambiguity 2) has been implicitly resolved, in this case as $a1 + c2 > a2 + c1$ or $a1 + c2 = a2 + c1$. Hence it is of no further use and can be discarded.

If instead the answer to Question 2 is *Yes* then it must be inferred that as well as it being theoretically possible for the decision makers, if they wanted to, to explicitly resolve the ambiguity in question as LHS > RHS (here $a1 + c2 > a2 + c1$) from Question 1 that it is

also theoretically possible for them to resolve it as RHS > LHS ($a2 + c1 > a1 + c2$). This implies that the ambiguity in question (here ambiguity 2) is *not* implicitly resolved as a corollary of the explicitly resolved ambiguities, and therefore it must be explicitly resolved by the decision makers.

In the case of ambiguity (2) $a1 + c2$ *vs* $a2 + c1$ the answers to Questions 1 and 2 are *Yes* and *Yes*, and so the ambiguity should be presented to the decision makers for them to explicitly resolve. As for the earlier demonstration of Approach 1, suppose it is decided $a1 + c2 > a2 + c1$.

The next ambiguity on the list — (3) $a2 + b1$ *vs* $a1 + b2$ — is then tested via the same process as outlined above. This time, though, the list of (actual) explicitly resolved inequalities/equalities against which (3) $a2 + b1 > a1 + b2$ (i.e., LHS > RHS, as for Question 1) and then, if necessary, (3) $a1 + b2 > a2 + b1$ (RHS > LHS, as for Question 2) are tested comprises (2) $a1 + c2 > a2 + c1$ as well as (1) $b2 + c1 > b1 + c2$ (as before).

Thus the list of explicitly resolved inequalities/equalities is continually updated — including in general but not in the present example, as for Approach 1, the identification of any explicitly resolved inequalities/equalities that are themselves corollaries of others on the list.

The process is repeated for all ambiguities until all of them have been identified as having been implicitly resolved or they are explicitly resolved by the decision makers. This can be summarised for the present example as follows.

For ambiguity (3) the answer to Question 1 is *No*, and therefore this ambiguity is identified as having been implicitly resolved as a corollary of the explicitly resolved ambiguities. For ambiguity (4), the answer to Question 1 is *Yes* but the answer to Question 2 is *No*, and therefore this ambiguity is identified as having been implicitly resolved.

For ambiguity (5) the answers to both questions are *Yes*, implying that this ambiguity has not been implicitly resolved, and so it must be explicitly resolved by the decision makers: as $a2 + b1 + c2 > a1 + b2 + c1$, as for the demonstration of Approach 1. Finally, for ambiguity (6) the answers are *Yes* and *No*, and therefore this ambiguity is also identified as having been implicitly resolved.

Thus, of the six ambiguities, three had to be explicitly resolved by the decision makers — (1) $b2 + c1 > b1 + c2$, (2) $a1 + c2 > a2 + c1$ and (5) $a2 + b1 + c2 > a1 + b2 + c1$ (the same three as for Approach 1) — with the other three (3, 4 and 6) identified as having been implicitly resolved as corollaries.

The final step of the method (Step 3) of the invention involves simultaneously solving the system of (independent) explicitly resolved ambiguities (inequalities and equalities) and inherent inequalities to obtain the point values. This step of the method is represented in Figure 5 at 530.

As described above, any explicitly resolved ambiguities that are themselves corollaries of other explicitly decided inequalities are removed, because only independent inequalities/equalities are required for deriving the point values.

For the exemplar APS with $x = 3$ and $y = 2$, one solution for (1) $b2 + c1 > b1 + c2$, (2) $a1 + c2 > a2 + c1$ and (5) $a2 + b1 + c2 > a1 + b2 + c1$ and $a2 > a1$, $b2 > b1$ and $c2 > c1$ is: $a1 = 0$, $a2 = 2$, $b1 = 0$, $b2 = 4$, $c1 = 0$ and $c2 = 3$. These point values produce ranking #9 in Figure 4.

The method explained above may be summarised as follows. First (Step 1), generate the ambiguities of the APS that is to be calibrated as shown at 510 in Figure 5.

Then (Step 2) *explicitly* resolve them via the value judgements of the consulted decision makers, while identifying all other ambiguities that are *implicitly* resolved as corollaries, until all ambiguities are resolved. This step is illustrated at 520 in Figure 5. Although, in

theory, the implicitly resolved ambiguities can be identified via *Approach 1* or *Approach 2* (both explained above), the latter is more efficient and therefore it is the particularly preferred method for the computer program of the invention, as explained below.

Finally (Step 3), simultaneously solve the system of explicitly resolved ambiguities (that is, inequalities and equalities) and inherent inequalities to obtain the point values for the APS. This step is illustrated at 530 in Figure 5.

The property (assumption) that the profile rankings of the decision makers are transitive (or logically consistent), enables the number of ambiguities that must be explicitly resolved at Step 2 to be minimised, with the remainder emerging implicitly as corollaries.

For APSs with more criteria and categories than the exemplar with $x = 3$ and $y = 2$, minimising the number of explicitly resolved ambiguities, and accordingly the number of value judgements required from the decision makers, is a significant practical advantage of our method.

The computer program for implementing the method described above will now be described more particularly. The computer program of the invention comprises, in broad terms, five steps, as explained in turn below. An overview of the five steps is illustrated in Figure 11.

Before starting the program, the user — who may be one of the decision makers consulted to resolve the ambiguities, or, alternatively, he or she may be a facilitator of the calibration process — must have chosen the criteria and categories of the APS that is to be calibrated and ranked each criterion's categories.

At Step 1, when the program begins, the user may be asked to enter a title for the APS and the criteria and their categories, and to rank the categories for each criterion. The criteria and categories must be labelled in terms of the variables (e.g. '$a1$', '$a2$' '$a3$', etc.) and verbally described in preparation for their later presentation to the decision makers.

The user is given the opportunity of listing theoretically impossible combinations of criteria and categories that partially or fully specify profiles — known as the *To Be-Excluded* list — and that are therefore to be used to cull ambiguities that are generated by the program. This list may be left empty if the user wishes.

After the program is initialised, it calculates the number of unique ambiguities to be resolved using equation (3) above. This can be reported to the user and used to estimate whether the system can be solved in an acceptable amount of time given the computing resources that are available.

Step 2 of the program involves generating the ambiguities using the 'efficient ambiguities generator' described earlier. An overview of this step is illustrated in Figure 12.

For simplicity and efficiency, ambiguities are generated one degree at a time, beginning with the 2^{nd}-degree, as determined by the value of a control variable — known as the *Current Degree* variable — which is initially set to 2.

As each ambiguity is generated it is checked for whether or not it is theoretically possible, given the *To Be Excluded* list of partially or fully specified profiles. If the ambiguity is theoretically possible it is then tested via Approach 2 of Step 2 of the method of the invention explained earlier for whether or not it is implicitly resolved by the explicitly resolved ambiguities. An overview of the procedure is illustrated in Figure 13.

This and similar tests may be performed via linear programming, which is described below.

If an ambiguity is found to be implicitly resolved then it is discarded; otherwise it is added to a list known as the *To Be Resolved* list. (Note that when the 2^{nd}-degree

ambiguities are generated, none will be implicitly resolved because none have yet been explicitly resolved.)

Step 3 of the program is to present the *To Be Resolved* list of unresolved ambiguities for the current degree to the user. An overview of this step and Step 4 (explained below) is illustrated in Figure 14.

The user can choose to view the ambiguities either in equation form (for example, $a1 + b2$ *vs* $a4 + b1$) or symbolically (for example, 21_ _ _ or 41_ _ _), which are ordered either randomly or by their criteria and categories. The user is invited to select an ambiguity for the purpose of explicitly resolving it. The selected ambiguity is 'translated' verbally in terms of the criteria and category descriptions.

If the user desires, he or she may skip this particular ambiguity and select another one, or she may resolve it by clicking one of three buttons labelled (in essence): "LHS greater" or "RHS greater" (i.e., < or RHS preferred) or "LHS and RHS equal". If "RHS greater (preferred)" is chosen, for consistency, the LHS and RHS of the resolved ambiguity are switched and stored as RHS > LHS.

The system can also permit weak inequalities (such as "LHS greater than or equal to" and "RHS greater than or equal to"), however this means that some ambiguities will later be partially (weakly) solved and so some buttons in the user interface must be disabled when the ambiguity is selected. Note that in any resulting APS the result will be either "greater than" or it will be "equal to" but it will not be both. The system can also be designed so that only strong inequalities and no equalities are permitted, thereby producing strict profile rankings only.

Step 4 is for the program to remove the explicitly resolved ambiguity (as above) from the *To Be Resolved* list and add it — as either an inequality or equality (depending on how the ambiguity was resolved) — to the list of explicitly resolved ambiguities, known as the *Explicitly Resolved* list. An overview of this step and Step 3 is illustrated as Figure 14.

Each inequality/equality on the *Explicitly Resolved* list is then tested as to whether or not it is implied by the others on the list. Any found to be implied may be marked as being 'redundant' and hereinafter ignored, but not deleted because they may be re-used later if any explicitly resolved ambiguities are later revised by the decision makers (explained below).

All ambiguities on the *To Be Resolved* list are then tested for whether or not their resolution is implied by the (non-redundant) inequalities/equalities on the *Explicitly Resolved* list, and if so they are deleted from the *To Be Resolved* list. This test was explained in earlier in terms of Hypothetical Questions 1 and 2 of Approach 2 of the method's Step 2 and is illustrated in Figure 13.

In essence, the test involves finding whether or not a solution (in terms of feasible point values) exists to a system comprising the explicitly resolved inequalities/equalities and inherent inequalities, and each of the possible hypothetical inequalities in turn (that is LHS > RHS and RHS > LHS) corresponding to the ambiguity in question. This and the earlier tests based on determining the existence of solutions are performed via linear programming (LP) with inequality and equality constraints. LP is discussed in more detail in the final section below.

If the *To Be Resolved* list is not empty, the program returns to Step 3. If instead the *To Be Resolved* list is empty and the *Current Degree* is equal to the number of criteria (x) in the particular APS being calibrated, the program proceeds to Step 5. Alternatively, if the *Current Degree* is less than the APS's number of criteria (and the *To Be Resolved* list is empty), the *Current Degree* is increased by '1' and the program returns to Step 2.

The user may choose at any time to view the explicitly resolved ambiguities for all degrees and 'undo' any of them. Explicitly resolved ambiguities that are 'redundant' are displayed in a different colour to alert the user to the fact that undoing them alone will

have no material effect. An overview of the 'undo module', as referred to in Figure 11, is illustrated in Figure 15.

When explicitly resolved ambiguities are undone, they are deleted from the *Explicitly Resolved* list. If any non-redundant explicitly resolved ambiguities are undone, the explicitly resolved ambiguities marked redundant are re-tested for whether or not they are still redundant, and marked accordingly.

Also, if non-redundant ambiguities are undone, the *Current Degree* variable is reset to the degree of the lowest-degree ambiguity that was undone, and the program restarts from the second step in the computer program as set out above.

When all the ambiguities for all degrees have been resolved, the list of explicitly resolved inequalities/equalities and inherent inequalities is solved via linear programming for the point values of the APS. These are presented to users, as well as a range of 'summary statistics', including the numbers of ambiguities and explicitly resolved ambiguities at each degree and, it the user wants them, the inequalities/equalities that were chosen.

In the interests of deriving point values that are integers and as low as possible, thereby maximising their 'user-friendliness', integer programming and an objective function that minimises the sum of the variables (for example, $a1 + a2 + a3 + b1 + b2 + b3 + c1 + c2 + c3$, etc.) may be specified.

Alternatively, were it desired by users that the point values be normalised to a particular range for the profile total scores, such as 0 to 100, the derived point values are scaled accordingly. However this usually forces some point values to be fractions, which is not as 'user-friendly' as integers. Accordingly both integer and normalised values can be calculated and presented.

As explained above, linear programming (LP) may be used in the computer program of the invention. Specifically, LP may be used at Steps 2 and 4 to determine whether the

resolution of an ambiguity is implied by the inequalities/equalities on the *Explicitly Resolved* list. In Step 5, LP is used for deriving the point values when all ambiguities have been resolved. The following features must be observed when LP is used.

First, the inequalities/equalities and inherent inequalities must be converted to a form suitable for LP. For example, the inherent inequalities $a3 > a2 > a1$ must be written as $a2 - a1 = 1$ and $a3 - a2 = 1$, and the explicitly resolved ambiguities $a2 + b2 = a3 + b1$ and $a2 + b3 > a3 + b2$ as $a2 + b2 - a3 - b1 = 0$ and $a2 + b3 - a3 - b2 = 1$. Setting the RHS of the weak inequality to "1" (that is, 'epsilon') corresponds to the initial inequality being strict, although other values for epsilon may perform at least as well.

Second, in Steps 2 and 4, the LP objective function need only be "0", as all that is being tested for is the *existence* of a solution rather than a particular optimal solution.

Finally, because the variables corresponding to the lowest category on the respective criteria ($a1$, $b1$, $c1$, and so on) are, effectively, 'numeraires' or 'baseline' values for the respective criteria, they can be set equal to zero, thereby eliminating them from the system to be solved and increasing the efficiency of the LP algorithm.

Below is an illustration of how LP may be used to test whether or not an ambiguity on the *To Be Resolved* list is implied by the inequalities/equalities on the *Explicitly Resolved* list.

For the sake of the example, suppose the ambiguity in question is $a1 + b3 + c3$ *vs* $a2 + b1 + c1$ and the following inequalities/equalities are on the *Explicitly Resolved* list (where the shaded inequality signifies that it is 'redundant' in the sense, as discussed earlier, that it is implied by others on the list).

$$\begin{gathered}
a2 + b2 = a3 + b1 \\
a2 + b3 > a3 + b2 \\
a3 + c1 > a1 + c3 \\
a2 + c1 > a1 + c3 \\
b1 + c2 = b3 + c1 \\
b1 + c3 = b3 + c2 \\
a1 + b2 + c3 > a2 + b1 + c1
\end{gathered}$$

In addition, the inherent inequalities of this exemplar APS with $x = 3$ and $y = 3$ are:

$$a3 > a2 > a1$$
$$b3 > b2 > b1$$
$$c3 > c2 > c1$$

These inequalities/equalities (and ignoring the shaded redundant inequality) — and given $a1 = b1 = c1 = 0$ (as discussed above) — are represented in a form suitable for LP, as follows. By definition, a solution (in terms of the point values) exists to the LP problem:

$$\begin{array}{rl} & \text{minimise } 0 \\ \text{subject to:} & a2 + b2 - a3 = 0 \\ & a2 + b3 - a3 - b2 \geq 1 \\ & a2 - c3 \geq 1 \\ & c2 - b3 = 0 \\ & c3 - b3 - c2 = 0 \\ & b2 + c3 - a2 \geq 1 \\ & a2 \geq 1 \\ & a3 - a2 \geq 1 \\ & b2 \geq 1 \\ & b3 - b2 \geq 1 \\ & c2 \geq 1 \\ & c3 - c2 \geq 1 \end{array}$$

To test whether $a1 + b3 + c3$ *vs* $a2 + b1 + c1$ is implied by the inequalities/equalities on the *Explicitly Resolved* list, the above LP problem is first augmented with $b3 + c3 - a2 \geq 1$ (i.e., $a1 + b3 + c3 > a2 + b1 + c1$), and tested for whether or not a solution to this new problem exists.

A solution does exist, and so the original LP problem is next augmented with $a2 - b3 - c3 \geq 1$ (i.e., $a2 + b1 + c1 > a1 + b3 + c3$) and tested for whether or not a solution to this second new problem exists.

In this case there is no solution, and so it must be inferred that either $a1 + b3 + c3 > a2 + b1 + c1$ or $a1 + b3 + c3 = a2 + b1 + c1$ is implied by the inequalities/equalities on the *Explicitly Resolved* list. The first inequality above is easily confirmed here by adding $b3 > b2$ to the last explicitly resolved ambiguity on the first list above: $(b3 > b2) + (a1 + b2$

$+ c3 > a2 + b1 + c1) = (a1 + b3 + c3 > a2 + b1 + c1)$. This alternative approach is variant 1 of Approach 1 explained earlier.

Therefore the ambiguity is implicitly resolved and deleted from the *To Be Resolved* list. (This process is performed for all the ambiguities on the *To Be Resolved* list.)

The foregoing describes the invention including preferred forms thereof. Alterations and modifications as will be obvious to those skilled in the art are intended to be incorporated within the scope hereof.

Paul Hansen & Franz Ombler

By the authorised agents
AJ PARK
Per

Western Canada Waiting List Project

HIP AND KNEE REPLACEMENT PRIORITY CRITERIA

PLEASE PRINT CLEARLY

Provincial Name: ______________________

Patient Age: __________ Sex: [circle one] M F

[Tick one box] ☐ Left Hip ☐ Right Hip ☐ Left Knee ☐ Right Knee

[Tick one box] ☐ Primary ☐ Revision

Diagnosis: ______________________

____

Surgeon's Name: ______________ Phone: ______________

Date of Patient Evaluation: ______________________

Patients must be on appropriate non-surgical treatment prior to evaluation (e.g. medications, walking aids, shoe inserts)

PLEASE CHECK THE BOX THAT MOST ACCURATELY DESCRIBES THE PATIENT'S CURRENT SITUATION

1. Pain on motion (e.g. walking, bending):*
   - ☐ None/Mild
   - ☐ Moderate
   - ☐ Severe
2. Pain at rest (e.g. while sitting, lying down or causing sleep disturbance):*
   - ☐ None
   - ☐ Mild
   - ☐ Moderate
   - ☐ Severe

*Take into account usual duration, intensity, and frequency of pain, including need for narcotic vs. non-narcotic medication

3. Ability to walk:
   - ☐ Over 5 blocks
   - ☐ 1-5 blocks
   - ☐ <1 blocks
   - ☐ Household ambulator
4. Other functional limitations (e.g. putting on shoes, managing stairs, sitting to standing, sexual activity, bathing, cooking, recreation or hobbies):
   - ☐ No limitations
   - ☐ Mild limitations (able to do most activities with minor modifications or difficulty)
   - ☐ Moderate limitations (able to do most activities but with modification or assistance)
   - ☐ Severe limitations (unable to perform most activities)

FIGURE 1

5. Abnormal findings on physical exam related to affected joint (e.g. deformity, instability, leg length difference, restriction of range of motion on examination):
   - ☐ None/Mild
   - ☐ Moderate
   - ☐ Severe

6. Potential for progression of disease documented by radiographic findings (e.g. recurrent dislocation, x-ray evidence of protrusion, significant bone loss, component wear, impending fracture):**
   - ☐ None
   - ☐ Mild
   - ☐ Moderate
   - ☐ Severe

** Predominantly applies to revisions, use in primary cases only in special circumstances (e.g. ligament instability, bone loss)

7. Threat to patient role and independence in society (e.g. ability to work, give care to dependents, live independently (difficulty must be related to affected joint)):
   - ☐ Not threatened but more difficult
   - ☐ Threatened but not immediately
   - ☐ Immediately threatened or unable

8. All things considered, how would you rate the urgency or relative priority of this patient? (Draw a line across the scale)

   Not Urgent at all —————————————————— Extremely Urgent (just short of an emergency)

9. In your clinical judgement, what should be the maximum waiting time for this patient?
   Number of weeks ____________ OR Number of months ____________

10. In your practice how long would it take this patient to have surgery done from the time you first see the patient?
   Number of days ________ OR Number of weeks ________ OR Number of months________

Please record any comments on the form, criteria or process: ____________________

____

____

____

____

____

FIGURE 2

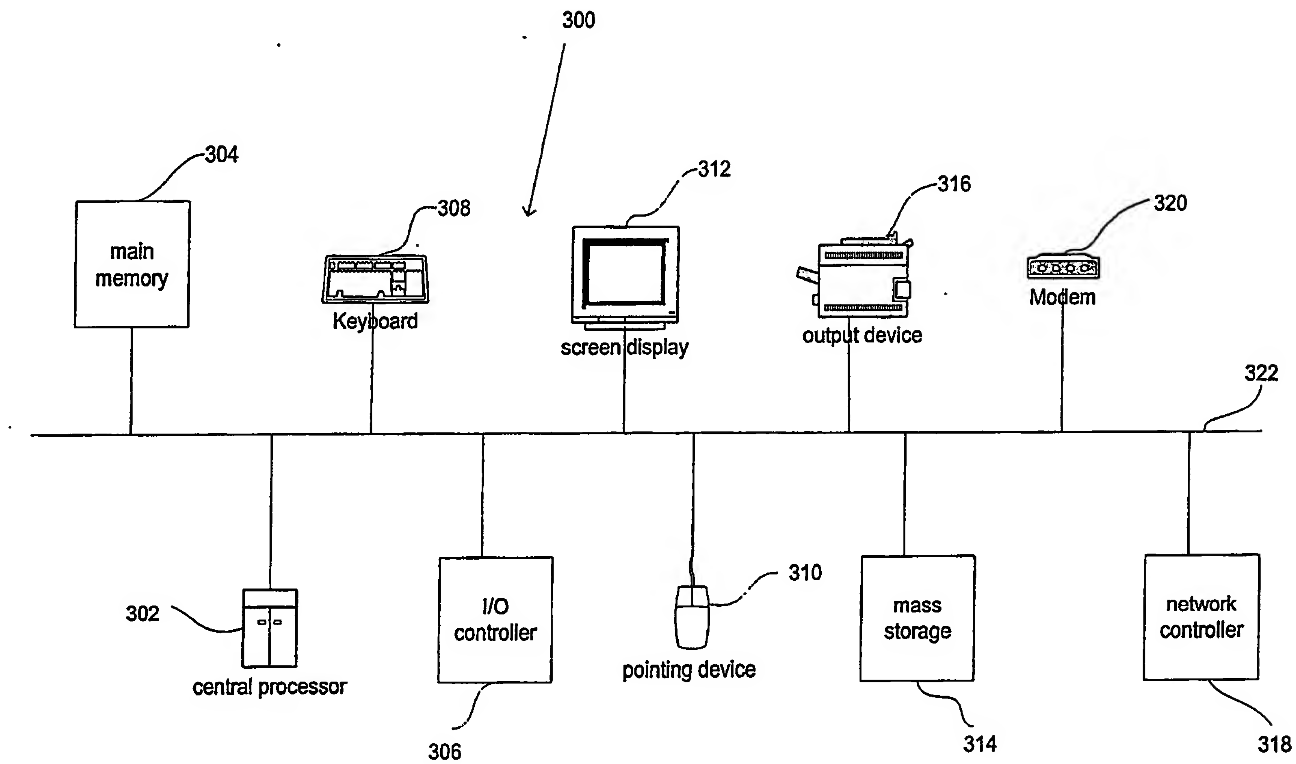
300
304
main memory
308
Keyboard
312
screen display
316
output device
320
Modem
322
302
central processor
I/O controller
306
310
pointing device
mass storage
314
network controller
318

FIGURE 3

```
                ? 211 ? 122 ? 121 ? 112 ? 111      (Ranking #1)
            ? 212
            ?   ? 122 ? 211 ? 121 ? 112 ? 111      (Ranking #2)
      ?  221?
      ?     ?   ? 212 ? 121 ? 211 ? 112 ? 111      (Ranking #3)
      ?     ? 122
      ?         ? 121 ? 212 ? 211 ? 112 ? 111      (Ranking #4)
      ?
      ?         ? 211 ? 122 ? 112 ? 121 ? 111      (Ranking #5)
      ?     ? 221
      ?     ?   ? 122 ? 211 ? 112 ? 121 ? 111      (Ranking #6)
  222 ?  212?
      ?     ?   ? 221 ? 112 ? 211 ? 121 ? 111      (Ranking #7)
      ?     ? 122
      ?         ? 112 ? 221 ? 211 ? 121 ? 111      (Ranking #8)
      ?
      ?         ? 212 ? 121 ? 112 ? 211 ? 111      (Ranking #9)
      ?     ? 221
      ?     ?   ? 121 ? 212 ? 112 ? 211 ? 111      (Ranking #10)
      ?  122?
            ?   ? 221 ? 112 ? 121 ? 211 ? 111      (Ranking #11)
            ? 212
                ? 112 ? 221 ? 121 ? 211 ? 111      (Ranking #12)
```

FIGURE 4

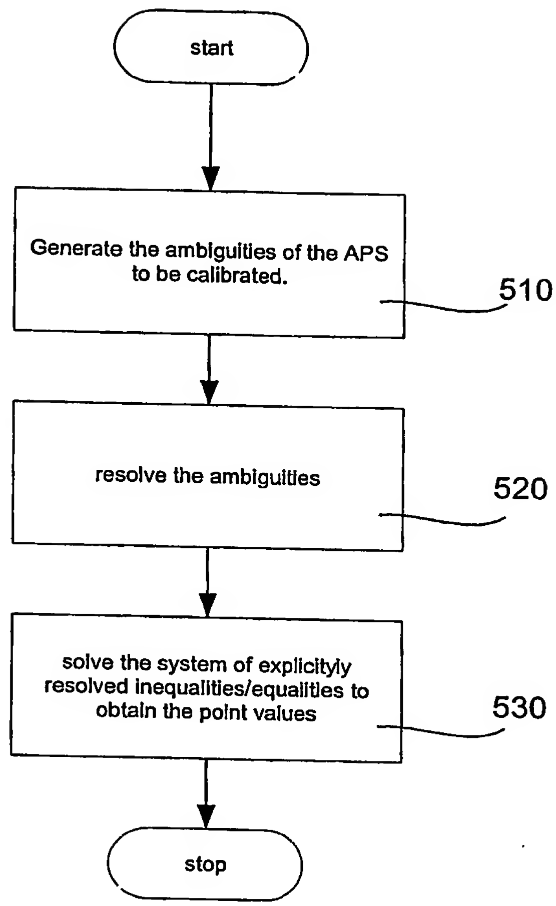
start
Generate the ambiguities of the APS to be calibrated.
510
resolve the ambiguities
520
solve the system of explicityly resolved inequalities/equalities to obtain the point values
530
stop

FIGURE 5

versus (*vs*)	221 *a2+b2+c1*	212 *a2+b1+c2*	122 *a1+b2+c2*	112 *a1+b1+c2*	121 *a1+b2+c1*	211 *a2+b1+c1*
221 *a2+b2+c1*		*b2+c1* *vs* *b1+c2*	*a2+c1* *vs* *a1+c2*	*a2+b2+c1* *vs* *a1+b1+c2*	n.a.	n.a.
212 *a2+b1+c2*			*a2+b1* *vs* *a1+b2*	n.a.	*a2+b1+c2* *vs* *a1+b2+c1*	n.a.
122 *a1+b2+c2*				n.a.	n.a.	*a1+b2+c2* *vs* *a2+b1+c1*
112 *a1+b1+c2*					*b1+c2* *vs* *b2+c1*	*a1+c2* *vs* *a2+c1*
121 *a1+b2+c1*						*a1+b2* *vs* *a2+b1*
211 *a2+b1+c1*						

FIGURE 6

Criteria (x)	Categories (y)	Profiles (y^x)	Pairwise comparisons $y^x(y^x - 1)/2$	Degrees	Total ambiguities (Equation 1)	Unique ambiguities (Equation 2)
3	2	8	28	2nd	6	3
				3rd	3	3
				All degrees:	*9*	*6*
3	3	27	351	2nd	81	27
				3rd	81	81
				All degrees:	*162*	*108*
3	4	64	2,016	2nd	432	108
				3rd	648	648
				All degrees:	*1,080*	*756*
4	2	16	120	2nd	24	6
				3rd	24	12
				4th	7	7
				All degrees:	*55*	*25*
4	3	81	3,240	2nd	486	54
				3rd	972	324
				4th	567	567
				All degrees:	*2,025*	*945*
4	4	256	32,640	2nd	3,456	216
				3rd	10,368	2,592
				4th	9,072	9,072
				All degrees:	*22,896*	*11,880*
4	5	625	195,000	2nd	15,000	600
				3rd	60,000	12,000
				4th	70,000	70,000
				All degrees:	*145,000*	*82,600*
5	2	32	496	2nd	80	10
				3rd	120	30
				4th	70	35
				5th	15	15
				All degrees:	*285*	*90*
5	3	243	29,403	2nd	2,430	90
				3rd	7,290	810
				4th	8,505	2,835
				5th	3,645	3,645
				All degrees:	*21,870*	*7,380*

FIGURE 7

Criteria (x)	Categories (y)	Profiles (y^x)	Pairwise comparisons $y^x(y^x-1)/2$	Degrees	Total ambiguities (Equation 1)	Unique ambiguities (Equation 2)
5	4	1,024	523,776	2nd	23,040	360
				3rd	103,680	6,480
				4th	181,440	45,360
				5th	116,640	116,640
				All degrees:	*424,800*	*168,840*
5	5	3,125	4,881,250	2nd	125,000	1,000
				3rd	750,000	30,000
				4th	1,750,000	350,000
				5th	1,500,000	1,500,000
				All degrees:	*4,125,000*	*1,881,000*
6	2	64	2,016	2nd	240	15
				3rd	480	60
				4th	420	105
				5th	180	90
				6th	31	31
				All degrees:	*1,351*	*301*
6	3	729	265,356	2nd	10,935	135
				3rd	43,740	1,620
				4th	76,545	8,505
				5th	65,610	21,870
				6th	22,599	22,599
				All degrees:	*219,429*	*54,729*
6	4	4,096	8,386,560	2nd	138,240	540
				3rd	829,440	12,960
				4th	2,177,280	136,080
				5th	2,799,360	699,840
				6th	1,446,336	1,446,336
				All degrees:	*7,390,656*	*2,295,756*
10	4	1,048,576	549,755,289,600	2nd	106,168,320	1,620
				3rd	1,274,019,840	77,760
				4th	7,803,371,520	1,905,120
				5th	30,098,718,720	29,393,280
				6th	77,755,023,360	303,730,560
				7th	135,444,234,240	2,116,316,160
				8th	153,584,087,040	9,599,005,440
				9th	102,792,499,200	25,698,124,800
				10th	30,898,215,936	30,898,215,936
				All degrees:	*539,756,338,176*	*68,646,770,676*

FIGURE 8

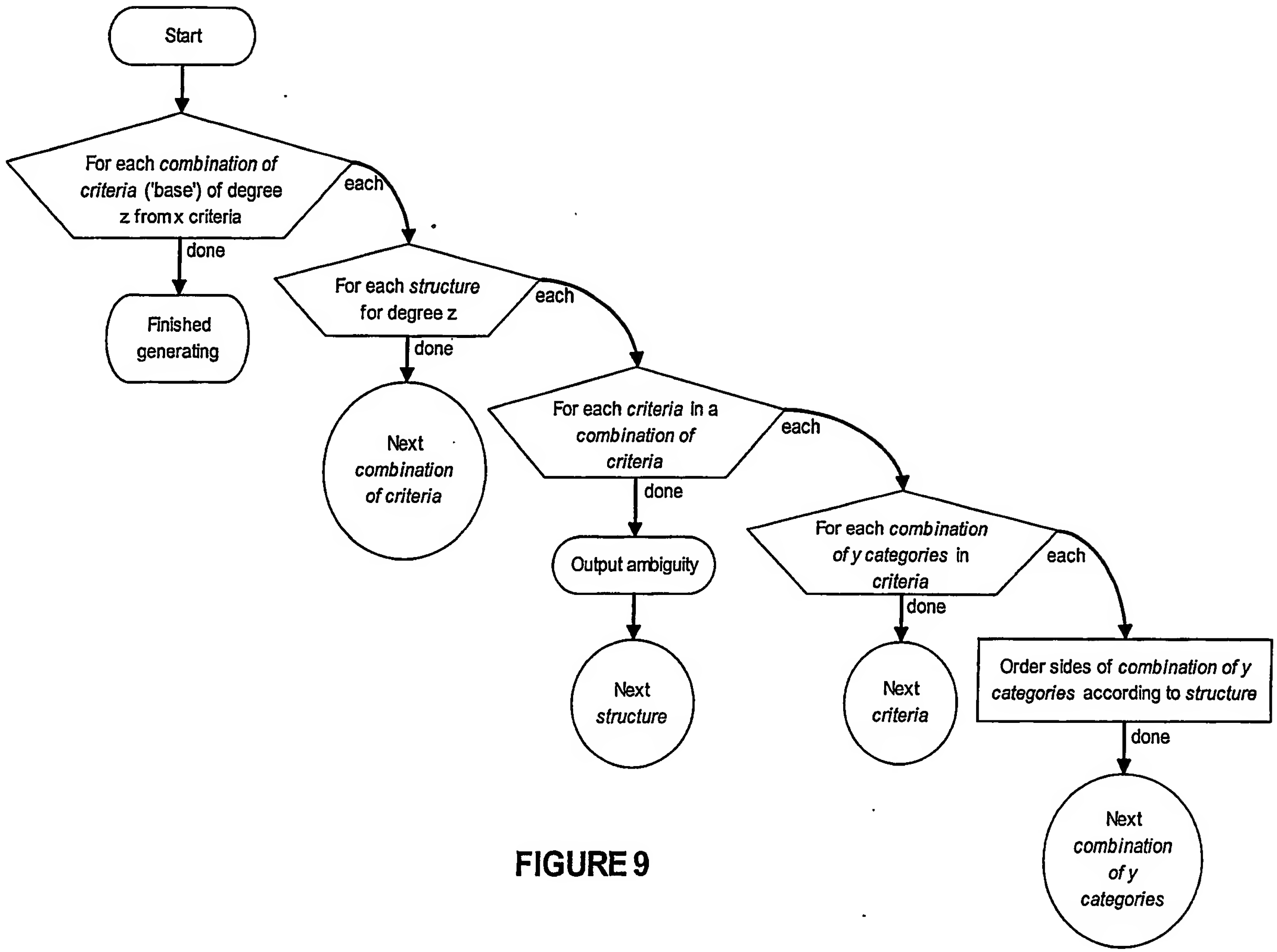
Start
For each *combination of criteria* ('base') of degree z from x criteria
each
done
Finished generating
For each *structure* for degree z
each
done
Next *combination of criteria*
For each *criteria* in a *combination of criteria*
each
done
Output ambiguity
Next *structure*
For each *combination of y categories* in *criteria*
each
done
Next *criteria*
Order sides of *combination of y categories* according to *structure*
done
Next *combination of y categories*

FIGURE 9

3^{rd}-degree ambiguity	Sufficient conditions for LHS > RHS of the 3^{rd}-degree ambiguity
(4) $a2 + b2 + c1$ *vs* $a1 + b1 + c2$	$a2 + c1 > a1 + c2$ or $a2 + c1 = a1 + c2$ or $b2 + c1 > b1 + c2$ or $b2 + c1 = b1 + c2$
(5) $a2 + b1 + c2$ *vs* $a1 + b2 + c1$	$b1 + c2 > b2 + c1$ or $b1 + c2 > b2 + c1$ or $a2 + b1 = a1 + b2$ or $a2 + b1 = a1 + b2$
(6) $a1 + b2 + c2$ *vs* $a2 + b1 + c1$	$a1 + c2 > a2 + c1$ or $a1 + c2 = a2 + c1$ or $a1 + b2 > a2 + b1$ or $a1 + b2 = a2 + b1$

FIGURE 10

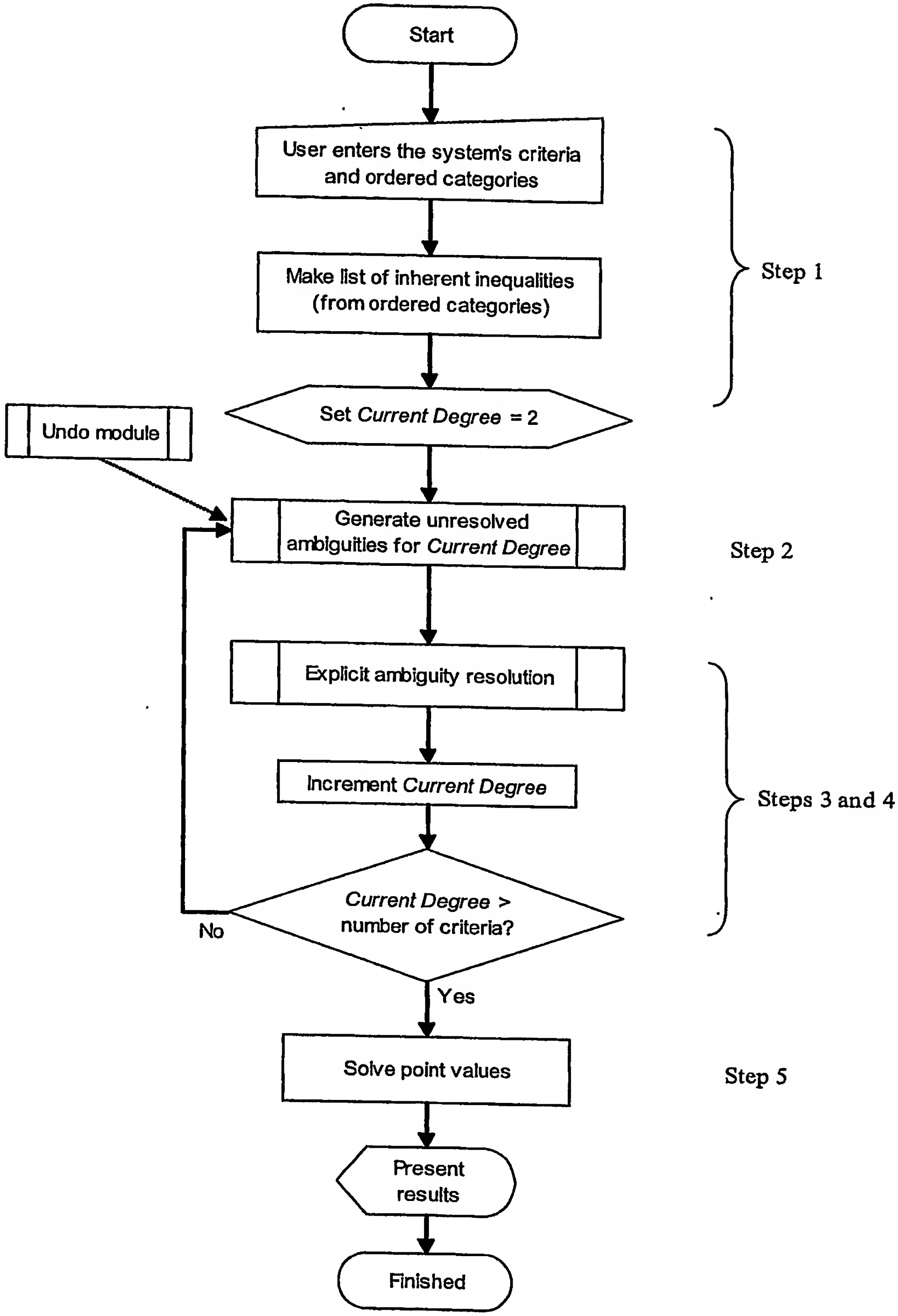

FIGURE 11

58423-1

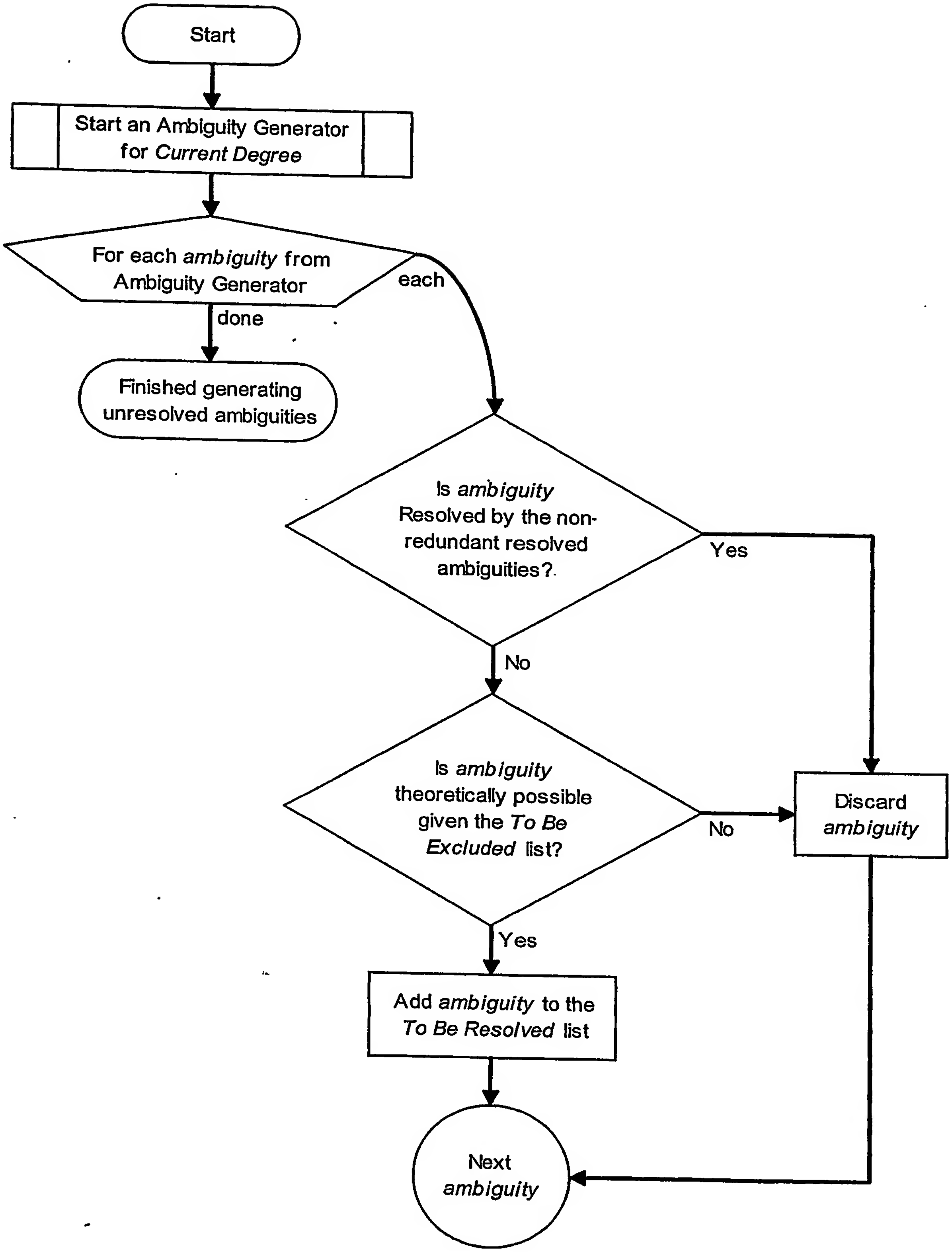

FIGURE 12

58474-1

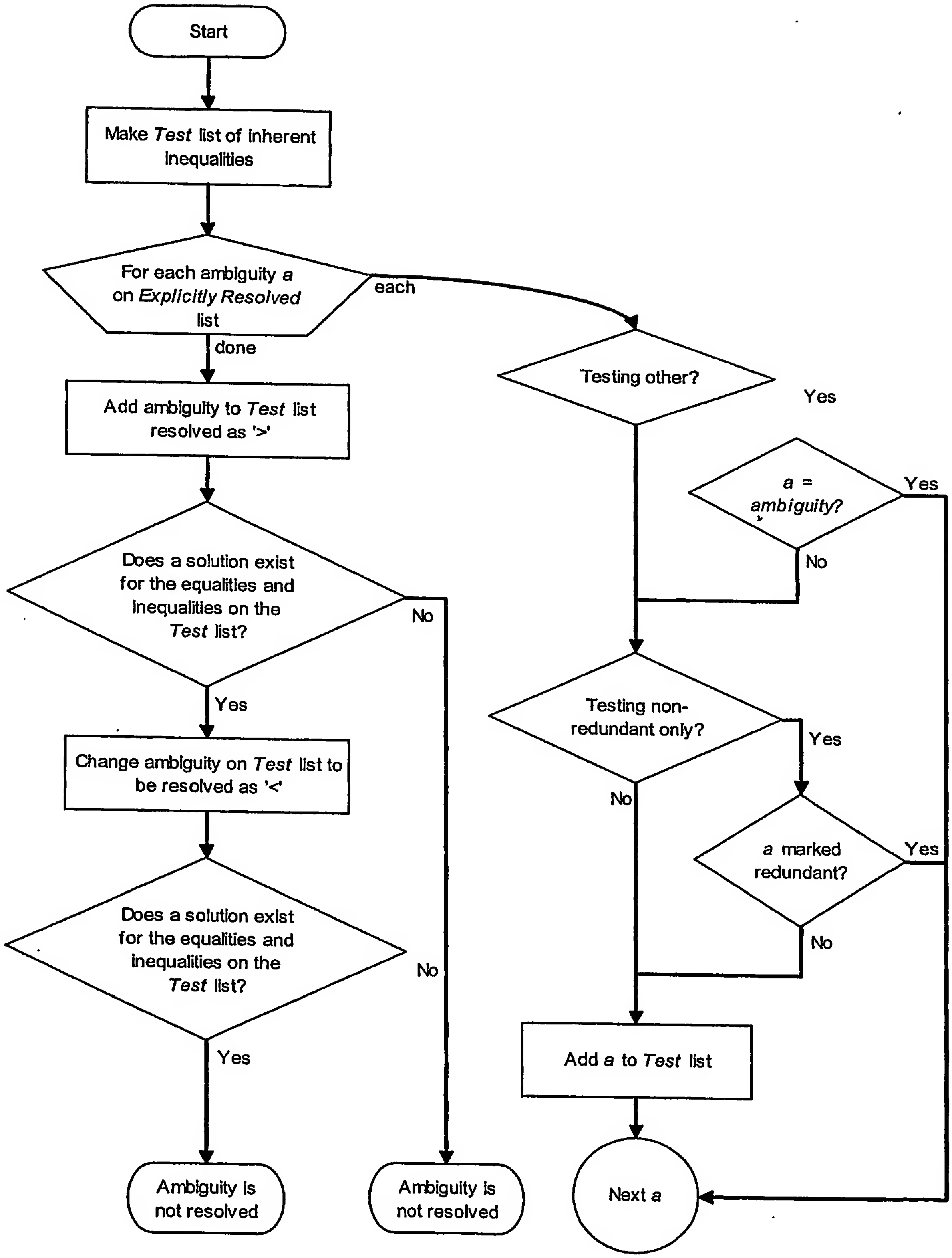

FIGURE 13

58477-1

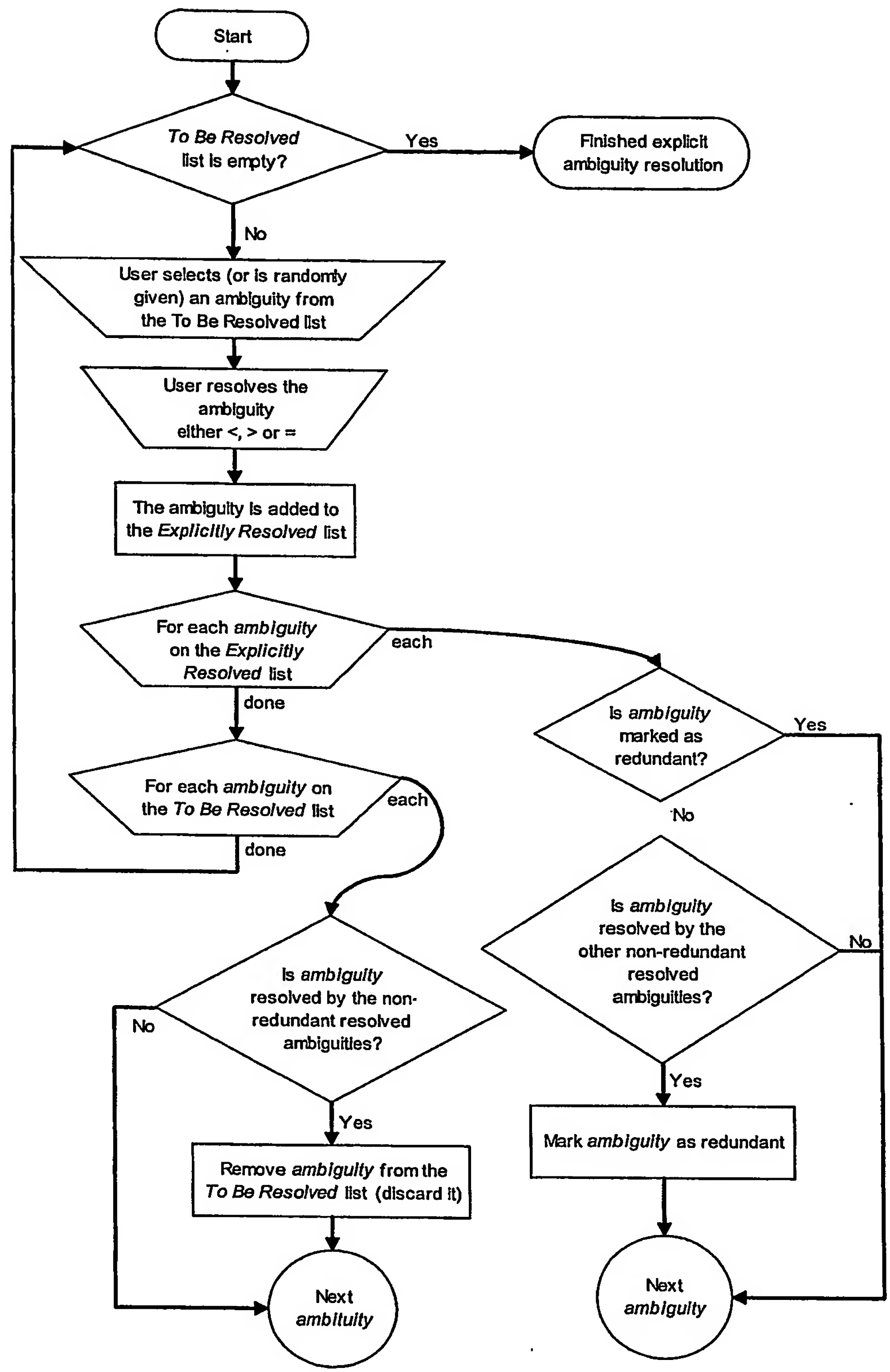

FIGURE 14

58502-1

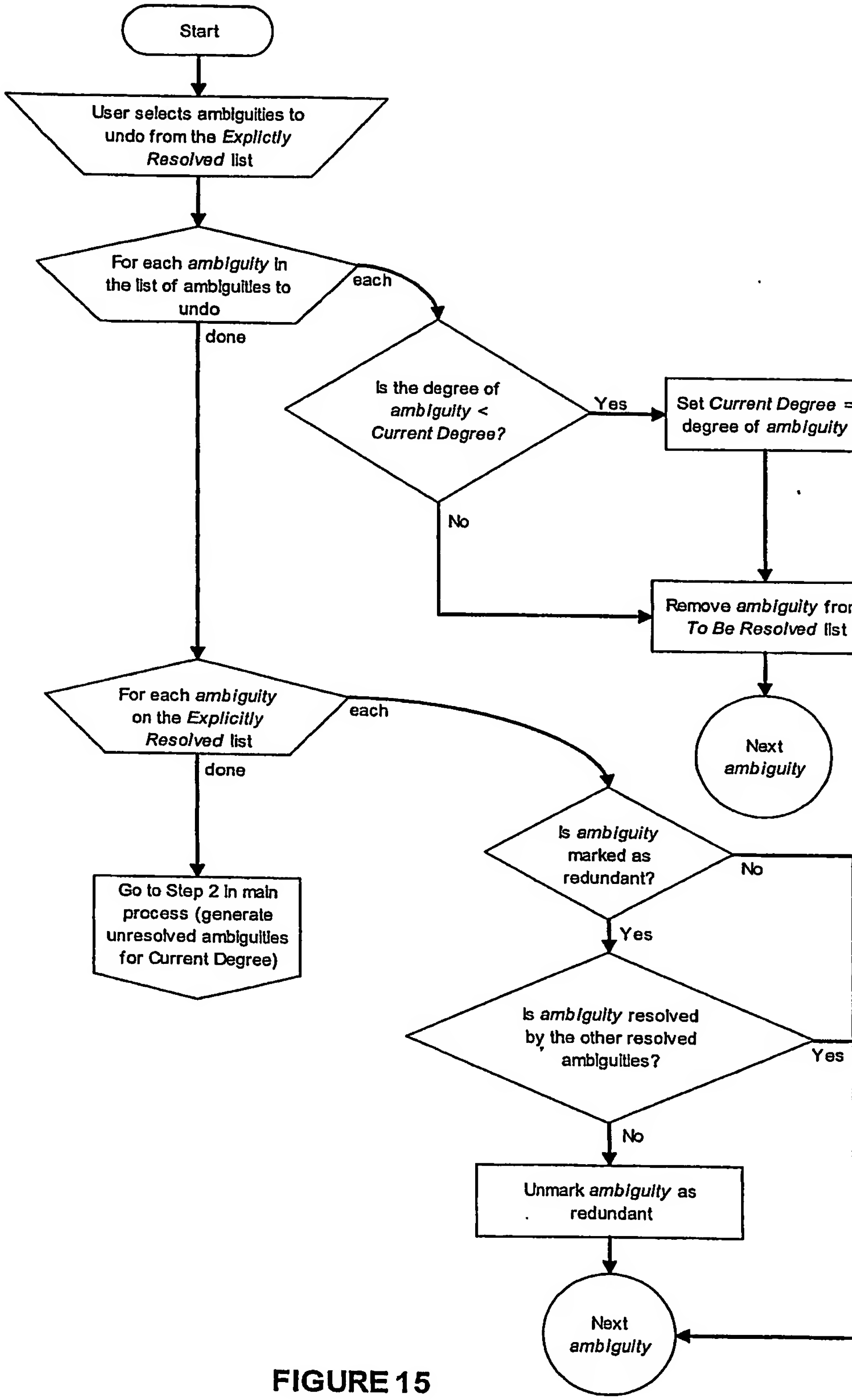
Start
User selects ambiguities to undo from the *Explictly Resolved* list
For each *ambiguity* in the list of ambiguities to undo
each
done
Is the degree of *ambiguity* < *Current Degree?*
Yes
No
Set *Current Degree* = degree of *ambiguity*
Remove *ambiguity* from *To Be Resolved* list
Next *ambiguity*
For each *ambiguity* on the *Explicitly Resolved* list
each
done
Go to Step 2 in main process (generate unresolved ambiguities for Current Degree)
Is *ambiguity* marked as redundant?
No
Yes
Is *ambiguity* resolved by the other resolved ambiguities?
Yes
No
Unmark *ambiguity* as redundant
Next *ambiguity*

FIGURE 15

Ministry of Economic Development
Manatū Ōhanga
Intellectual Property Office of New Zealand

PCT/NZ2004/000121

CERTIFICATE

This certificate is issued in support of an application for Patent registration in a country outside New Zealand pursuant to the Patents Act 1953 and the Regulations thereunder.

I hereby certify that annexed is a true copy of the Provisional Specification as filed on 25 August 2003 with an application for Letters Patent number 527785 made by PAUL HANSEN and FRANZ OMBLER.

Dated 7 July 2004.

Neville Harris

Neville Harris
Commissioner of Patents, Trade Marks and Designs

COMMISSIONER OF PATENTS, TRADEMARKS & DESIGNS
NEW ZEALAND

330 High Street, Lower Hutt and 17 Toop Street, Seaview
PO Box 30687, Lower Hutt, New Zealand or DX SX 11129, Wellington
Phone: +64 4 560 1600, Fax: +64 4 560 1691, www.iponz.govt.nz

527785

Patents Form No. 4

PATENTS ACT 1953

PROVISIONAL SPECIFICATION

A CHOICE-BASED METHOD FOR CHOOSING FROM A RESTRICTED GROUP OF ALTERNATIVES

We, **PAUL HANSEN, a New Zealand citizen, of 24 Maitland Street, Dunedin, New Zealand, and FRANZ OMBLER, a New Zealand citizen, of 19 Edinburgh Terrace, Berhampore, Wellington, New Zealand,** do hereby declare this invention to be described in the following statement:

INTELLECTUAL PROPERTY OFFICE OF N.Z.
25 AUG 2003
RECEIVED

A CHOICE-BASED METHOD FOR CHOOSING FROM A RESTRICTED GROUP OF ALTERNATIVES

FIELD OF INVENTION

The invention relates to the field of Decision Analysis or Decision Support and more particularly to a method for choosing an alternative (the 'best' alternative) from a restricted group of alternatives, and/or ranking those alternatives.

BACKGROUND TO THE INVENTION

The invention is an extension of an earlier invention of the applicants: "A CHOICE-BASED METHOD FOR CALIBRATING ADDITIVE POINTS SYSTEMS", for which provisional patent application NZ 526647 was filed on 13 June 2003. A full copy of this patent application is included in the Appendix to this specification and is hereby incorporated into the present disclosure. This earlier invention comprises a system, method and computer program for determining (calibrating) the points values of additive points systems (APSs).

The background to the invention described in the disclosure of the Appendix is broadly applicable to the present invention. However, supplementary to the situations in which APSs are applicable are situations in which decision makers are required to choose an alternative (the 'best' alternative) from a restricted group of alternatives, and/or to rank those alternatives. Decision makers in a wide variety of settings are regularly faced with having to do this; examples include choosing: the best location for a building, the best model of car to buy, the best employee to hire or promote, etc.

Although the invention described in the Appendix is capable of choosing the 'best' alternative from a restricted group of alternatives, and/or ranking them, it is designed primarily for calibrating APSs in general. As such the Appendix describes the ranking of very large groups of alternatives, thereby, in effect, ranking all theoretically possible alternatives, including alternatives that are as yet unknown or are purely hypothetical.

The present invention extends the invention described in the Appendix such that, instead of having to rank all theoretically possible alternatives, the present invention ranks a subset of them such that the 'best' alternative is chosen from a restricted group of alternatives and/or they are ranked. This serves to reduce the amount of effort required of decision makers in reaching their decision.

SUMMARY OF THE INVENTION

In broad terms, the present invention extends the system, method and computer program described in the Appendix so that the 'best' alternative is chosen from a restricted group of alternatives, and/or they are ranked.

The invention does this by extending the invention described in the Appendix to explicitly recognise that in the process of calibrating a given APS, the restricted group of alternatives may have been ranked *before* the APS is fully calibrated. In the language of this earlier invention, the restricted group of alternatives may have been ranked by the explicitly resolved ambiguities together with the inherent inequalities before the set of all theoretically possible profiles of the APS has been ranked.

In broad terms in one form therefore, the invention provides a method for ranking a restricted group of alternatives in the context of an additive points system (APS) comprising a pre-determined plurality of criteria, each criterion having one or more categories wherein the points for each category of each criterion are determined by the pairwise ranking of profile pairs, each profile comprising a set of one or more criteria, each criterion in the set instantiated with one of the categories for that criterion wherein the restricted group of alternatives comprises two or more of these sets.

Preferably the method comprises the steps of generating ambiguous profile pairs for the APS; simultaneously generating all possible ambiguities that are consistent with the restricted group of alternatives and storing them on a temporary list; making a second temporary list that initially consists of the alternatives from the restricted group of

alternatives; solving the system of equalities/inequalities for the APS to obtain point values; removing ambiguities from the temporary list as the corresponding ambiguities for the APS are solved; removing alternatives from the second temporary list as they are found to be certainly ranked lower than another on the list; and returning a solution when the temporary list is empty or when the second temporary list contains equally ranked alternatives, depending on the needs of the decision makers.

Preferably the process of solving ambiguities for the APS may be halted once the temporary list is empty.

The invention also provides a related system and computer program implementing corresponding extensions to the system and computer program described in the Appendix.

The invention therefore extends the system, method and computer program disclosed in the Appendix by determining the exact point at which the restricted group of alternatives has been ranked.

DETAILED DESCRIPTION OF THE PREFERRED FORMS

This description extends the detailed description of the preferred forms of the Appendix, and uses the same terms and refers to the figures contained and described in the Appendix.

The invention is primarily embodied in the methodology set out there both by itself and as implemented through computing resources such as the preferred resources set out in Figure 3 of the Appendix, by way of example. The invention is also embodied in the software used to implement the methodology and in any system comprising a combination of hardware and software used to implement the methodology.

The present description begins with reference to Step 1 of the description of the computer program for implementing the method described at page 37, line 28 of the Appendix. Thus when the user enters the criteria and categories of the APS being calibrated (as per Figure 11 of the Appendix), the present invention invites him or her to also list the restricted group of alternatives in terms of their characteristics with respect to the APS's criteria and categories.

The method of the present invention will now be explained with reference to the example of ranking the following four alternatives in the case of an APS with just three criteria: *a*, *b* and *c*; and two categories on each: 1 and 2; such that there are six criterion-category variables: $a1$, $a2$, $b1$, $b2$, $c1$ and $c2$. By definition, the values of these variables monotonically increase with the categories within each criterion so that the following 'inherent inequalities' hold: $a2 > a1$, $b2 > b1$ and $c2 > c1$. This also is the first example of an APS referred to in the Appendix.

The four alternatives and their corresponding total score equations are (where the three-digit numbers are symbolic representations only):

$$\text{Alternative 1: } 221 = a2 + b2 + c1$$
$$\text{Alternative 2: } 122 = a1 + b2 + c2$$
$$\text{Alternative 3: } 212 = a2 + b1 + c2$$
$$\text{Alternative 4: } 211 = a2 + b1 + c1$$

The essence of ranking these four alternatives, and/or choosing the 'best' one, is deciding the point values of the six variables ($a1$, $a2$, $b1$, $b2$, $c1$ and $c2$) such that decision makers' preferred ranking of the four alternatives (equations) is realised. The system, method and computer program described in the Appendix is extended to do this as follows.

When the program begins to generate unresolved ambiguities for the *Current Degree* (as per Step 2 of Figure 11 of the Appendix), it also generates all the possible ambiguities

that are consistent with the restricted group of alternatives. These are stored on a list known as the *Alternatives' Ambiguities* list.

The method for generating these ambiguities is the same as described on page 13, lines 10 to 19 of the Appendix. In essence, it involves pairwise comparisons of alternatives and cancelling variables common to both alternatives.

Thus the *Alternatives' Ambiguities* list for the four alternatives in the example are:

$$a2 + c1 \text{ } vs \text{ } a1 + c2$$

$$b2 + c1 \text{ } vs \text{ } b1 + c2$$

$$a2 + b1 \text{ } vs \text{ } a1 + b2$$

$$a1 + b2 + c2 \text{ } vs \text{ } a2 + b1 + c1$$

Immediately before setting the *Current Degree* = 2 (Step 1 of Figure 11 of the Appendix) and in the Undo module, the restricted group of alternatives nominated by the user is copied to a list known as the *Remaining Alternatives* list.

During the normal processes described in the Appendix, after the user resolves the particular ambiguity being considered, and it is added to the *Explicitly Resolved* list (as per Figure 14 of the Appendix), the present invention operates as follows.

Ambiguities on the *Alternatives' Ambiguities* list are tested and removed in the same way as the *To Be Resolved* list (as per Figure 14 of the Appendix). If the *Alternatives' Ambiguities* list is not then empty, the process continues as per the process described in the Appendix; however, once the *Alternatives' Ambiguities* list is empty, all alternatives have been ranked.

The system then solves the point values (Step 5, Figure 11 of the Appendix) to obtain the scores for the alternatives, and then presents the ranked alternatives to the user. The user can continue to calibrate the APS if he or she wishes, but the ranking of the alternatives

will not change unless the user undoes some prior decisions or introduces new alternatives to the restricted group under consideration.

If the *Alternatives' Ambiguities* list is not yet empty, a test is performed as to whether there is a clear 'winner' or a set of equally ranked winners among the *Remaining Alternatives*. To perform this second test the *Remaining Alternatives* list is processed as follows.

Each alternative on the *Remaining Alternatives* list is compared with every other to determine whether it would be infeasible for it to be ranked below the other alternatives, given the *Explicitly Resolved* list and the inherent inequalities.

With respect to Alternatives 1 and 2, for example, the system asks two hypothetical questions in a similar manner to Approach 2 described at page 33, line 10 of the Appendix.

Hypothetical Question 1: Does a solution exist to the system comprising the (actual) explicitly resolved inequalities/equalities as well as the inherent inequalities and the proposition Alternative 1 ≥ Alternative 2 (here $a2 + b2 + c1 \geq a1 + b2 + c2$). If the answer is *No*, and therefore it would not be theoretically possible for Alternative 1 to be ranked above or equal to Alternative 2, then it must be true that Alternative 1 is ranked below Alternative 2. Hence Alternative 1 is removed from the *Remaining Alternatives* list.

If instead the answer to Question 1 is *Yes*, and therefore it would be theoretically possible for Alternative 1 to be ranked above or equal to Alternative 2, then the following second hypothetical question is asked.

Hypothetical Question 2: Does a solution exist to the system comprising the (actual) explicitly resolved inequalities/equalities as well as the inherent inequalities and the proposition Alternative 2 ≥ Alternative 1 (here $a1 + b2 + c2 \geq a2 + b2 + c1$). If the

answer is *No*, and therefore it would not be theoretically possible for Alternative 2 to be ranked above or equal to Alternative 1, then it must be true that Alternative 2 is ranked below Alternative 1. Hence Alternative 2 is removed from the *Remaining Alternatives* list.

If instead the answer to Question 2 is *Yes*, and therefore it would be theoretically possible for Alternative 1 to be ranked above or equal to Alternative 2, then the relationship between these two alternatives is either not yet known or they are equal.

After all the alternatives on the (now possibly shorter) *Remaining Alternatives* list are tested in this manner, a final test is performed pairwise comparing each alternative on the list with every other to determine if it is possible that they are not equal, in a fashion analogous to the above.

For example if it is possible that Alternative 1 < Alternative 2, or that Alternative 2 < Alternative 1, (i.e., by testing whether it is possible that here $a2 + b2 + c1 < a1 + b2 + c2$, or $a1 + b2 + c2 < a2 + b2 + c1$) then the comparison of alternatives is abandoned and the overall process described in the Appendix continues.

If instead all the alternatives are found to be equally ranked, or if there is only one, the user is presented with the 'winning' (i.e., 'best') alternative or alternatives. The user can continue to calibrate the APS if he or she wishes, but the highest ranking alternative or alternatives will not change unless the user undoes some prior decisions or introduces new alternatives to the restricted group under consideration.

The foregoing, taken together with the Appendix, describes the invention including preferred forms thereof. Alterations and modifications as will be obvious to those skilled in the art are intended to be incorporated within the scope hereof.

APPENDIX

A CHOICE-BASED METHOD FOR CALIBRATING ADDITIVE POINTS SYSTEMS

FIELD OF INVENTION

The invention relates to additive points systems and in particular to a system, method and computer program for determining (calibrating) the points values of additive points systems.

BACKGROUND TO THE INVENTION

Additive points systems (APSs), a type of multi-attribute utility model or multiple criteria decision analysis tool, are also known as 'linear', 'point-count' and 'scoring' systems. APSs are widely and increasingly used worldwide in most branches of medicine for prioritisation, diagnosis and predictive purposes and in a wide range of other applications, including selecting immigrants, assessing mortgage applications and predicting parole violations, business bankruptcies and college graduations. They are also used as a generic project appraisal tool.

Canada and New Zealand, for example, recently developed APSs for a wide range of elective surgeries and other publicly-funded health cares. APSs are also used by the immigration systems of New Zealand, Canada, Australia and Germany and, to a lesser extent, the United Kingdom.

APSs in general represent a relatively simple solution to the pervasive problem faced by decision makers with multiple criteria or attributes to consider, particularly when ranking alternatives or individuals. Hereinafter in this document alternatives and individuals are referred to generically as 'alternatives'.

Specifically, APSs serve to combine alternatives' characteristics on *multiple* criteria to produce a *single* ranking of alternatives with respect to an over-arching criterion (such as, for example, the order in which to treat patients), or, more simply, to reach a decision (for example, whether or not to admit an immigrant).

In addition to their having been near universally found in many studies to be more accurate than 'expert' decision makers in the respective fields to which they have been applied, the appeal of points systems is that they are simple to use.

Each alternative's categorical rating on each criterion that is deemed relevant to the over-arching criterion is scored a particular number of points, that usually increase with the 'importance' of the categories, and the points are *summed* (hence *additive* points systems) to produce a total score for the alternative.

Alternatives are ranked with respect to the over-arching criterion according to their total scores, including being declined altogether if a particular score 'threshold' is not reached. Usually the higher an alternative's score the higher its ranking, and the scores typically have no other meaning than this.

Most APSs for elective surgeries and immigration respectively (the examples referred to above), have between five and seven criteria and two to five categories on each criterion. Figures 1 and 2 show an example of an APS used in Canada for prioritising patients for hip or knee replacement surgery.

In this example there are seven criteria, mostly based on types of pain and functional limitations. Each criterion has a number of mutually exclusive and exhaustive categories on which the consulted decision maker (usually a doctor) is asked to rate the patient being considered. For example, item 2 in Figure 1, refers to the criterion of "Pain at Rest" and requires that the patient be assigned to one of four categories: "None", "Mild", "Moderate" or "Severe".

In general, if the criteria and the categories on each have been chosen for a particular APS system, then the point values for that APS must be determined (calibrated) such that the resulting ranking of alternatives represents the decision makers' preferences. The invention is a new system, method and computer program for doing this.

In addition to the arbitrary assignment of points, there are two main existing approaches to calibrating APSs.

The first regresses decision makers' judgements of the relative priorities or importance of a sample of real or hypothetical alternative 'profiles' (in other words, the alternative's categorical ratings on the criteria) on their characteristics in terms of the criteria and derives point values from the regression coefficients. Usually only a small proportion of existing or theoretically possible profiles is surveyed because of the responder burden on the consulted decision makers.

The above-mentioned decision makers' judgements are often elicited via a visual analogue scale (VAS). Item 8 of Figure 2 is such a VAS, where the decision maker is asked "to rate the urgency or relative priority of this patient" between "Not urgent at all" and "Extremely urgent (just short of an emergency)". This 'score' and the patient's characteristics in terms of Items 1 to 7 of the same figure, along with analogous data for other patients, may then be used to calibrate the point values for the various criteria and categories using multiple regression techniques, as explained above.

The second existing approach to calibration uses decision makers' judgements of the pairwise relative importance of the APS's criteria to derive ratio scale weights. These weights are then applied to normalised criteria values to derive point values. An example of this type of technique is the Analytic Hierarchical Process (AHP).

Thus the first approach assumes that decision makers' judgements have *interval scale* measurement properties and the second approach assumes they have *ratio scale* properties. Both assumptions are relatively stringent and current techniques for eliciting decision makers' judgements have well-known biases. With respect to the first approach, for example, the validity of the dependent variable (the experts' judgments) and therefore the point values derived from the estimated coefficients, can be criticised on two main grounds.

First, the scaling methods such as the VAS used to elicit the experts' judgments of the profiles' relative priorities are based on mere introspection rather than the expression of a choice. Second, VAS in general may not actually have the scaling measurement properties required for the valuations that they produce to be interpreted as relative priorities rather than just as rankings.

It is therefore desirable to have a method of calibrating new APSs or recalibrating or validating extant ones that requires only *ordinal* measurement properties, specifically the positive expression of a ranking over pairs of alternatives, as this is the least stringent of measurement property requirements. It would also be desirable for this method to achieve accurate results while reducing the burden on decision makers of ranking pairs of alternatives by minimising the number of pairs they have to rank.

SUMMARY OF THE INVENTION

In broad terms, in one form the invention provides a method for calibrating additive points systems (APSs) comprising a pre-determined plurality of criteria, each criterion having one or more categories wherein the points for each category of each criterion are determined by the pairwise ranking of profile pairs, each profile comprising a set of one or more of the criteria, each criterion in the set instantiated with one of the categories for that criterion.

Preferably the method comprises the steps of generating ambiguous profile pairs for the APS to be calibrated, resolving the ambiguous profile pairs, and solving the resulting system of equalities/inequalities to obtain the point values. Ambiguous profile pairs are profile pairs in which one profile has a *higher* categorical rating on at least one criterion and a *lower* rating on at least one other criterion than the other profile.

Preferably the step of generating the ambiguous profile pairs comprises the further step of removing any profiles that are theoretically impossible.

Preferably the step of generating the ambiguous profile pairs comprises the further step of reducing all profile pairs in which both profiles have one or more of the same criteria instantiated with the same category.

In broad terms in another form, the invention provides a system for calibrating additive points systems (APSs) comprising a pre-determined plurality of criteria for the additive points system, each criterion capable of being instantiated with one or more pre-defined categories; and a points calibrator configured to determine appropriate points for each category of each criterion by preparing data for and processing the results of the pairwise ranking of profile pairs, each profile comprising a set of one or more of the criteria, each criterion in the set instantiated with one of the categories for that criterion.

Preferably the points calibrator comprises an ambiguity generator configured to generate ambiguous profile pairs for the APS to be calibrated.

Preferably the points calibrator comprises a data input component configured to receive and store the equalities/inequalities that result from resolving the ambiguous profile pairs generated by the ambiguity generator.

Preferably the points calibrator comprises a solution component configured to solve the resulting system of equalities/inequalities to obtain the point values.

Preferably the ambiguity generator is further configured to remove any profiles that are theoretically impossible.

Preferably the ambiguity generator is further configured to reduce all profile pairs in which both profiles have one or more of the same criteria instantiated with the same category.

In broad terms in yet another form the invention provides a computer program for calibrating additive points systems (APSs) comprising an initialization component

configured to receive and store data representing a plurality of criteria for an APS and the categories with which each criterion may be instantiated, an ambiguity generator configured to generate ambiguous profile pairs for the APS to be calibrated, a resolution component configured to select and present profile pairs to a user to be explicitly resolved and to store the results of the resolution, an ambiguity management component configured to manage the resolved and unresolved ambiguities and to automatically resolve any ambiguities that can be resolved implicitly; and a solution component configured to solve the system of resolved inequalities/equalities from the resolution component and the ambiguity management component.

Preferably the computer program further comprises a revision component configured to allow a user to revise any resolved ambiguities.

Preferably the computer program is implemented using linear programming.

BRIEF DESCRIPTION OF THE FIGURES

Preferred forms of the method system and computer program for calibrating additive points systems will now be described with reference to the accompanying figures in which:

Figure 1 shows a prior art means of eliciting expert judgments for the purposes of calibrating an APS via multiple regression-based techniques;

Figure 2 shows a continuation of the prior art means of eliciting expert judgments for the purposes of calibrating an APS via multimedia regression-based techniques;

Figure 3 shows a preferred configuration of hardware for carrying out the invention;

Figure 4 shows a decision tree identifying the 12 rankings of the 8 profiles possible in an exemplar APS with three criteria and two categories and allowing strict preferences only and no ties (for illustrative purposes only);

Figure 5 is a flow diagram illustrating the main steps in the method and computer program of the invention;

Figure 6 is a table showing the ambiguities from the six profiles (excluding those that are unambiguous) for an exemplar APS with three criteria and two categories;

Figure 7 is a table of the total and unique ambiguities for different APSs and degrees;

Figure 8 is a continuation of the table from Figure 7 of the total and unique ambiguities for different APSs and degrees;

Figure 9 is a flow diagram illustrating the main components of the 'efficient ambiguities generator' described in Step 1 of the method of the invention;

Figure 10 is a table of the sufficient (but not necessary) conditions for implicitly resolving the 3rd-degree ambiguities of an exemplar APS with three criteria and two categories;

Figure 11 is a flow diagram illustrating an overview of Steps 1 to 5 of the computer program of the invention;

Figure 12 is a flow diagram illustrating the preferred main components of the ambiguity generator of the computer program;

Figure 13 is a flow diagram illustrating the preferred main components involved in testing whether or not an ambiguity is resolved for the computer program;

Figure 14 is a flow diagram illustrating the preferred main components involved in the explicit resolution of ambiguities for the computer program; and

Figure 15 is a flow diagram illustrating the preferred main components of the undo module (whereby explicitly resolved ambiguities are revised) of the computer program.

DETAILED DESCRIPTION OF THE PREFERRED FORMS

The invention is primarily embodied in the methodology set out below both by itself and as implemented through computing resources such as the preferred resources set out in Figure 3, by way of example. The invention is also embodied in the software used to implement the methodology and in any system comprising a combination of hardware and software used to implement the methodology.

In its most preferred form the invention is implemented on a personal computer or workstation operating under the control of appropriate operating and application software.

Figure 3 shows the preferred system architecture of a personal computer, workstation, or server on which the invention could be implemented. The computer system 300 typically comprises a central processor 302, a main memory 304, for example RAM, and an input/output controller 306. The computer system 300 may also comprise peripherals such as a keyboard 308, a pointing device 310, for example a mouse, touchpad, or trackball, a display or screen device 312, a mass storage memory 314, for example a hard disk, floppy disk or optical disc and an output device 316 such as a printer. The system 300 could also include a network interface card or controller 318 and/or a modem 320. The individual components of the system 300 could communicate through a system bus 322.

The method of the present invention will now be described with reference to several examples, beginning with an APS with just three criteria: *a*, *b* and *c*; and two categories on each: 1 and 2; such that there are six criterion-category variables: $a1$, $a2$, $b1$, $b2$, $c1$ and $c2$.

For example, if this were an APS for selecting immigrants, criterion *a* might be 'educational qualifications', *b* 'wealth' and *c* 'language proficiency'. Category 1 might be generically defined as 'low' and 2 as 'high'. Real immigration APSs, however, typically have at least twice as many criteria and categories as this simple example.

By definition, the values of these variables monotonically increase with the categories within each criterion so that the following 'inherent inequalities' hold: $a2 > a1$, $b2 > b1$ and $c2 > c1$.

Corresponding to all possible combinations of the two categories on the *abc* criteria, eight profiles, each with a total score equation, are represented by this system (where the three-digit numbers are symbolic representations only):

$$\begin{aligned}
222 &= a2 + b2 + c2 \\
221 &= a2 + b2 + c1 \\
212 &= a2 + b1 + c2 \\
122 &= a1 + b2 + c2 \\
112 &= a1 + b1 + c2 \\
121 &= a1 + b2 + c1 \\
211 &= a2 + b1 + c1 \\
111 &= a1 + b1 + c1
\end{aligned}$$

The essence of calibrating an APS such as this one is deciding the point values of the six variables ($a1$, $a2$, $b1$, $b2$, $c1$ and $c2$) such that decision makers' preferred (or 'valid') overall ranking of the eight profiles (equations) is realised.

The internal logic of APSs — specifically, the inviolable laws of arithmetic — restricts the otherwise $8! = 40{,}320$ rankings (permutations), given strict preferences only and no ties (for illustrative purposes only), to the 12 rankings represented via the decision tree in Figure 4. The decision tree highlights the inherent contingencies in the derivations of the profile's rankings.

Any of the 12 rankings shown in Figure 4 can be produced from the six variables, depending on the values chosen for them. Ranking #1, for example, is given by $a1 = 0$, $a2 = 4$, $b1 = 0$, $b2 = 2$, $c1 = 0$ and $c2 = 1$, with the total scores: $222 = 7$, $221 = 6$, $212 = 5$,

$211 = 4$, $122 = 3$, $121 = 2$, $112 = 1$ and $111 = 0$. Alternatively, ranking #12, for example, is given by $a1 = 0$, $a2 = 1$, $b1 = 0$, $b2 = 2$, $c1 = 0$ and $c2 = 4$, with the total scores: $222 = 7$, $122 = 6$, $212 = 5$, $112 = 4$, $221 = 3$, $121 = 2$, $211 = 1$ and $111 = 0$.

In general, any particular ranking of profiles is determined by their *pairwise* rankings vis-à-vis each other. For an APS with x criteria and y categories on each, and y^x profiles, a maximum of $\frac{y^x(y^x-1)}{2}$ pairwise rankings is possible. Thus for the exemplar APS with $x = 3$ and $y = 2$, and eight profiles (i.e., 2^3), there is a maximum of 28 [i.e., $8(8 - 1)/2$] pairwise rankings.

The method of the invention minimises the number of pairwise rankings that must be decided *explicitly* (via value judgements), such that, in this example, a minimum of two and a maximum of four, rather than 28, is required, with the remaining pairwise rankings *implicitly* resolved as corollaries of the explicit rankings. The method of the invention comprises, in broad terms, three steps, as explained in turn below. The three main steps of the invention are illustrated in Figure 5.

Step 1 of the method of the invention 510 involves identifying (or 'generating') the 'ambiguities' of the APS that is being calibrated.

Ambiguities are formed from the total score equations of profile pairs whose pairwise rankings are a priori *ambiguous*, given the inherent inequalities (as already explained, $a2 > a1$, $b2 > b1$ and $c2 > c1$ in the present example). These are profile pairs in which one profile has a *higher* categorical rating on at least one criterion and a *lower* rating on at least one other criterion than the other profile.

For example, the pairwise ranking of profiles 221 and 212 — hereinafter referred to as "221 *vs* 212" and corresponding to $a2 + b2 + c1$ *vs* $a2 + b1 + c2$ — is ambiguous given $a2 > a1$, $b2 > b1$ and $c2 > c1$, as noted above.

On the other hand, many profile pairs are unambiguously ranked. For example, 222 ($a2 + b2 + c2$) is always pairwise ranked first and 111 ($a1 + b1 + c1$) is always pairwise ranked second — e.g., $a2 + b2 + c2 > a1 + b2 + c2$ (122) and $a1 + b2 + c2$ (122) $> a1 + b1 + c1$, and so on. Other profile pairs are similarly unambiguously ranked, for example, $a1 + b2 + c2 > a1 + b1 + c2$ (122 > 112), and so on.

As described above, identifying and eliminating all possible unambiguous pairwise rankings serves to cull the rankings of the eight possible profiles from 8! permutations (40,320) to 48 (for illustrative purposes only, given strict preferences only and no ties) from which the 12 in Figure 4 are further culled via another aspect of the internal logic of APSs described below.

The method of the invention identifies and excludes profile pairs that are unambiguously ranked and then focuses exclusively on profile pairs that are ambiguously ranked.

Some ambiguously ranked profile pairs can be 'reduced' by cancelling variables that are common to both profile equations. Thus $a2 + b2 + c1$ *vs* $a2 + b1 + c2$ (221 *vs* 212, as above) can be reduced to $b2 + c1$ *vs* $b1 + c2$ by cancelling $a2$ from both profiles' equations. In effect, because $a2$ appears in both equations it has no bearing on the ranking of the two profiles that the equations represent. We refer to such reduced forms as 'ambiguities'.

Moreover $b2 + c1$ *vs* $b1 + c2$ also corresponds to 121 *vs* 112, as $a1 + b2 + c1$ *vs* $a1 + b1 + c2$ reduces to $b2 + c1$ *vs* $b1 + c2$ after cancelling the $a1$ terms from both profiles' equations. Thus $b2 + c1$ *vs* $b1 + c2$ represents *two* ambiguously ranked profile pairs: 121 *vs* 112 *and* 221 *vs* 212.

However, not all ambiguously ranked profile pairs are reducible in this fashion. For example, no variables can be cancelled from $a2 + b2 + c1$ *vs* $a1 + b1 + c2$ (221 *vs* 112), as none are common to both profiles' equations. Nonetheless, such irreducible ambiguously ranked profile pairs are also hereinafter referred to as ambiguities.

Accordingly ambiguities can be classified by the number of criteria they contain, hereinafter referred to as the 'degree' of the ambiguity. Thus '2nd-degree' ambiguities contain two criteria, for example, $b2 + c1$ *vs* $b1 + c2$, as above, and '3rd-degree' ambiguities contain three criteria, for example, $a2 + b2 + c1$ *vs* $a1 + b1 + c2$, as above, and so on.

In general, the ambiguities for an APS with x criteria range from 2nd-degree to x^{th}-degree.

The algorithmically simplest process for generating ambiguities is to first create all y^x combinations of the y categories on the x criteria (that is, all profiles), and then pairwise rank them all against each other to identify pairs in which one profile has a higher categorical rating on at least one criterion and a lower rating on at least one other criterion. There will always be a total of $\frac{y^x(y^x-1)}{2}$ pairwise rankings.

As each ambiguously ranked profile pair is identified, it is reduced where possible by canceling variables in both profiles' equations, and retained only if the resulting ambiguity has not already been discovered. In other words, replicated ambiguities are discarded.

Accordingly the ambiguities for the exemplar APS with $x = 3$ and $y = 2$ are set out in Figure 6, where the shaded ambiguities are duplicates of ones reported earlier in the table. In Figure 6, rankings that are unambiguous are denoted by 'n.a.'. The blank elements of the matrix, except for the main diagonal, are mirror images of the ones reported, and shaded ambiguities are duplicates of ones reported earlier.

Thus it can be seen in Figure 6 that although there are nine ambiguities in total, only six are unique:

(1) $b2 + c1$ *vs* $b1 + c2$
(2) $a2 + c1$ *vs* $a1 + c2$
(3) $a2 + b1$ *vs* $a1 + b2$

(4) $a2 + b2 + c1$ *vs* $a1 + b1 + c2$
(5) $a2 + b1 + c2$ *vs* $a1 + b2 + c1$
(6) $a1 + b2 + c2$ *vs* $a2 + b1 + c1$

Clearly ambiguities (1) to (3) are 2nd-degree ambiguities (and are duplicated in the figure) and ambiguities (4) to (6) are 3rd-degree ambiguities.

In general, the number of ambiguities of each degree can be calculated from the two equations described below.

For a given APS with x criteria and y categories on each criterion, the *total* number of ambiguities (including replicates) of a given degree (z) is given by:

$$\frac{x!}{(x-z)!z!} \times (2^{z-1} - 1) \times \left(\frac{y(y-1)}{2}\right)^{z} \times y^{x-z} \qquad (1)$$

The first term corresponds to ${}^{x}C_{z}$ (the combinations formula); the second term is the powers of 2 and minus one; the third term is ${}^{y}C_{2}$ raised to the z^{th}-power or, alternatively, the sum of the first $y - 1$ natural numbers (e.g., $5(5-1)/2 = 10 = 1 + 2 + 3 + 4$) raised to the z^{th}-power; and the last term is a power function.

Of these ambiguities, the number of *unique* ambiguities (excluding replicates) of a given degree is given by:

$$\frac{x!}{(x-z)!z!} \times (2^{z-1} - 1) \times \left(\frac{y(y-1)}{2}\right)^{z} \qquad (2)$$

The two equations (1) and (2) differ only by the term ($\times$) y^{x-z}. The meaning of this and the other three terms (common to both equations) is discussed further below when an alternative process for generating ambiguities is described.

Equations (1) and (2) can be compared with the equation mentioned earlier for the total number of pairwise rankings (including unambiguous ones): $\frac{y^{x}(y^{x}-1)}{2}$.

Equation (2) is particularly useful because it reveals how many ambiguities at each degree must be resolved for any given APS. Figures 7 and 8 report the numbers of total and unique ambiguities for different APSs and degrees. As illustrated there, the number of ambiguities can be relatively large even for small values of x and y. For example, an APS with $x = 6$ and $y = 4$ (twice the values for the exemplar APS above) has a total of 2,295,756 unique ambiguities across its five degrees (2^{nd} to 6^{th}).

The simple process described above for generating ambiguities is further simplified if any profiles that are theoretically impossible are culled from the set to be pairwise ranked before any ambiguities are generated.

For example, in the APS for hip or knee replacements referred to in Figures 1 and 2 it would be a contradiction for a patient to be rated as having "*severe*" "*Pain on motion (e.g., walking, bending)*" while also being rated on another of the criteria as having the "*Ability to walk without significant pain*" for a distance of "over 5 blocks". Such a combination of categories on these two criteria is theoretically impossible and therefore all profiles that include it could be deleted from the list to be pairwise ranked.

Similarly, when validating an extant APS (rather than calibrating a new one), the profiles to be evaluated can be determined from a 'stocktake' of the alternatives ranked by the APS over its lifetime. Any other profiles that might realistically be expected in the future could be added.

This process also serves to increase the efficiency of Steps 2 and 3 of the method of the invention described below, as in general the more profiles that are culled, the fewer ambiguities there are, and therefore the simpler is the calibration exercise.

Notwithstanding such refinements, the simple process described above is computationally inefficient because profile comparisons that are ultimately unnecessary are performed and replicated ambiguities are discarded.

For example, to generate the above-mentioned 2,295,756 ambiguities for an APS with $x = 6$ and $y = 4$, as can be calculated from the data in Figure 8, 6,090,804 unnecessary profile comparisons are performed and 5,094,900 ambiguities are discarded.

The particularly preferred process for generating ambiguities according to the invention is therefore described below. This is also the preferred process for the computer program of the invention. The main components of the process are illustrated in Figure 9.

The particularly preferred process, hereinafter referred to as the 'efficient ambiguities generator', is described with explicit reference to the three terms in equation (2) above (reproduced below) and a specific example. The example is that of generating the 3rd-degree ambiguities for an APS with $x = 5$ and $y = 3$, of which there are 810 in total (as can be calculated from equation (2) and is reported in Figure 7).

Although the equation applies to APSs with the same number of categories on the criteria, as explained later below, the process can be generalised to allow the number of categories to vary across criteria. For example, instead of $y = 3$ for all 5 criteria in the above-mentioned APS, as is applied later, criterion *a* could have two categories, *b* three categories, *c* four, and so on.

$$\frac{x!}{(x-z)!z!} \times (2^{z-1}-1) \times \left(\frac{y(y-1)}{2}\right)^z \qquad (2)$$

Equation (2)'s *first term* — $\frac{x!}{(x-z)!z!}$ ($= {}^xC_z$, the combinations formula) — is the number of combinations of z criteria that can be selected from x criteria.

For the above-mentioned APS with $x = 5$ (that is, criteria *a*, *b*, *c*, *d* and *e*), 10 combinations of three criteria ($z = 3$) are possible: *abc*, *abd*, *abe*, *acd*, *ace*, *ade*, *bcd*, *bce*, *bde* and *cde*. Well-known algorithms exist for generating such combinations.

Each combination can be thought of as forming the 'base' for a group of ambiguities. For example, *abc* is the base for all 3rd-degree ambiguities centred on $a + b + c$ *vs* $a + b + c$ (hereinafter abbreviated to *abc vs abc*), such as $a2 + b3 + c3$ *vs* $a3 + b2 + c2$.

Because the criteria in each base (10 of them in the present example) have the same number of categories each, the bases can be treated identically with respect to the following operations that correspond to the other two terms in equation (2).

The *second term* $(2^{z-1}-1)$ can be interpreted as representing the number of underlying 'structures' for a given degree (z). Structures are generated by first listing the numbers between 1 and $2^{z-1} - 1$ in binary form using z bits, where each bit represents a criterion of two categories represented by either 0 or 1.

Continuing with the example of generating the 3rd-degree ambiguities for an APS with $x = 5$, the structures for $z = 3$ are:

011
010
001

These structures correspond to 3, 2, and 1 in binary form. For now, each structure can be thought of as being analogous to a profile with $y = 2$ for all criteria. Therefore only one ambiguity can be created from each, which is done by finding each structure's '1s complement', that is, by 'flipping the bits'; thus in the example:

011 *vs* 100
010 *vs* 101
001 *vs* 110

Each term (either a '0' or a '1') on either side of the ambiguity structure represents a criterion, and for each criterion their relative magnitudes ('0' versus '1') represents the relative magnitude of the categories they represent (i.e., 'low' or 'high'), such that each structure has an underlying pattern.

Thus 011 *vs* 100 in the example above (and analogously for 010 *vs* 101 and 001 *vs* 110) signifies that the first of the three criteria represented (corresponding to 0_ _ *vs* 1_ _) has a *lower* category on the left hand side (LHS) and a *higher* category on the right hand side (RHS) of each ambiguity that is derived from it.

Similarly, the second and third criteria (corresponding to _11 *vs* _00) both have *higher* categories on the LHS and *lower* categories on the RHS.

The equation's *third term* — $\left(\frac{y(y-1)}{2}\right)^{z}$ — is the number of ambiguities that can be generated from each ambiguity structure, as determined by the number of categories (y) on the criteria in a given base. There are three steps to generating these ambiguities, as follows.

First, all of the bases are matched with all of the structures. In the present example, the 10 bases (*abc*, *abd*, *abe*, *acd*, *ace*, *ade*, *bcd*, *bce*, *bde* and *cde*) are matched with the three structures (011 *vs* 100, 010 *vs* 101 and 001 *vs* 110) to produce $10 \times 3 = 30$ matches:

abc vs abc with 011 *vs* 100
abc vs abc with 010 *vs* 101
abc vs abc with 001 *vs* 110
abd vs abd with 011 *vs* 100
... and so on for another 26 matches.

Second, the underlying 'pattern' for each match (that is, a base with a structure) is implemented, according to the number of categories on the bases.

For *abc vs abc* with 011 *vs* 100, for example, with three categories on the criteria ($y = 3$), there are *three* ways each (that is, $\frac{y(y-1)}{2}$ = 3(3 – 1)/2 ways) of representing criterion *a* with a *lower* category on the LHS and a *higher* category on the RHS (0_ _ *vs* 1_ _), and criteria *b* and *c* both with *higher* categories on the LHS and *lower* categories on the RHS (_11 *vs* _00):

*a*2 *vs a*3   *b*3 *vs b*2   *c*3 *vs c*2

$a1$ *vs* $a3$	$b3$ *vs* $b1$	$c3$ *vs* $c1$
$a1$ *vs* $a2$	$b2$ *vs* $b1$	$c2$ *vs* $c1$

Note that $\frac{y(y-1)}{2} = {}^{y}C_2$, the number of combinations of categories (within the criterion) taken two-at-a-time. The output listed above embodies these combinations, where the two categories are simply ordered as required by the structure.

Finally, all $3 \times 3 \times 3 = 27$ possible combinations — i.e., $\left(\frac{y(y-1)}{2}\right)^z = [3(3-1)/2]^3$ — of these three patterns are formed, thereby obtaining all 27 3rd-degree ambiguities corresponding to the match *abc vs abc* with 011 *vs* 100:

$a2 + b3 + c3$ *vs* $a3 + b2 + c2$
$a2 + b3 + c3$ *vs* $a3 + b2 + c1$
$a2 + b3 + c2$ *vs* $a3 + b2 + c1$
$a2 + b3 + c3$ *vs* $a3 + b1 + c2$
$a2 + b3 + c3$ *vs* $a3 + b1 + c1$
$a2 + b3 + c2$ *vs* $a3 + b1 + c1$
$a2 + b2 + c3$ *vs* $a3 + b1 + c2$
$a2 + b2 + c3$ *vs* $a3 + b1 + c1$
$a2 + b2 + c2$ *vs* $a3 + b1 + c1$
$a1 + b3 + c3$ *vs* $a3 + b2 + c2$
$a1 + b3 + c3$ *vs* $a3 + b2 + c1$
$a1 + b3 + c2$ *vs* $a3 + b2 + c1$
$a1 + b3 + c3$ *vs* $a3 + b1 + c2$
$a1 + b3 + c3$ *vs* $a3 + b1 + c1$
$a1 + b3 + c2$ *vs* $a3 + b1 + c1$
$a1 + b2 + c3$ *vs* $a3 + b1 + c2$
$a1 + b2 + c3$ *vs* $a3 + b1 + c1$
$a1 + b2 + c2$ *vs* $a3 + b1 + c1$
$a1 + b3 + c3$ *vs* $a2 + b2 + c2$
$a1 + b3 + c3$ *vs* $a2 + b2 + c1$
$a1 + b3 + c2$ *vs* $a2 + b2 + c1$
$a1 + b3 + c3$ *vs* $a2 + b1 + c2$
$a1 + b3 + c3$ *vs* $a2 + b1 + c1$
$a1 + b3 + c2$ *vs* $a2 + b1 + c1$
$a1 + b2 + c3$ *vs* $a2 + b1 + c2$
$a1 + b2 + c3$ *vs* $a2 + b1 + c1$
$a1 + b2 + c2$ *vs* $a2 + b1 + c1$

Analogues of the three steps explained above are performed for all matches. For each of the 30 matches in the present example with $x = 5$, $y = 3$ and $z = 3$, 27 ambiguities analogous to the 27 above are generated, resulting in a total of 30 × 27 = 810 (unique) 3rd-degree ambiguities — i.e., $\frac{x!}{(x-z)!z!} \times (2^{z-1} - 1) \times \left(\frac{y(y-1)}{2}\right)^z = 10 \times 3 \times 27 = 810$.

As noted earlier, the process outlined above can be generalised to allow the number of categories to vary across the criteria. The key difference from the process explained above is that the underlying 'pattern' for each match (for example, *abc vs abc* with 011 *vs* 100) is idiosyncratic to the criteria included, as determined by the numbers of categories on each criterion.

For example, if in the example referred to above, instead of three categories on all criteria, criterion *a* has two categories, *b* has three and *c* four. Thus with just two categories for criterion *a* (for 011 *vs* 100, corresponding to 0_ _ *vs* 1_ _), there is only one underlying pattern ($\frac{y(y-1)}{2}$ = 2(2 – 1)/2) corresponding to a *lower* category on the LHS and a *higher* category on the RHS of each ambiguity that is derived:

$a1$ *vs* $a2$

As before, with three categories for criterion *b* (corresponding to _1_ *vs* _0_), there are three underlying patterns ($\frac{y(y-1)}{2}$ = 3(3 – 1)/2) corresponding to a *higher* category on the LHS and a *lower* category on the RHS:

$b3$ *vs* $b2$
$b3$ *vs* $b1$
$b2$ *vs* $b1$

Finally, with four categories for criterion *c* (corresponding to _ _1 *vs* _ _0), there are six underlying patterns ($\frac{y(y-1)}{2}$ = 4(4 – 1)/2) corresponding to a *higher* category on the LHS and a *lower* category on the RHS:

$c4$ *vs* $c3$

$$c4 \ vs \ c2$$
$$c4 \ vs \ c1$$
$$c3 \ vs \ c2$$
$$c3 \ vs \ c1$$
$$c2 \ vs \ c1$$

By taking all $1 \times 3 \times 6 = 18$ combinations of the above three sets of criteria-categories, all 18 3rd-degree ambiguities corresponding to *abc vs abc* with 011 *vs* 100 (with two, three and four categories respectively) may be obtained:

$$a1 + b3 + c4 \ vs \ a2 + b2 + c3$$
$$a1 + b3 + c4 \ vs \ a2 + b2 + c2$$
$$a1 + b3 + c4 \ vs \ a2 + b2 + c1$$
$$a1 + b3 + c3 \ vs \ a2 + b2 + c2$$
$$a1 + b3 + c3 \ vs \ a2 + b2 + c1$$
$$a1 + b3 + c2 \ vs \ a2 + b2 + c1$$
$$a1 + b3 + c4 \ vs \ a2 + b1 + c3$$
$$a1 + b3 + c4 \ vs \ a2 + b1 + c2$$
$$a1 + b3 + c4 \ vs \ a2 + b1 + c1$$
$$a1 + b3 + c3 \ vs \ a2 + b1 + c2$$
$$a1 + b3 + c3 \ vs \ a2 + b1 + c1$$
$$a1 + b3 + c2 \ vs \ a2 + b1 + c1$$
$$a1 + b2 + c4 \ vs \ a2 + b1 + c3$$
$$a1 + b2 + c4 \ vs \ a2 + b1 + c2$$
$$a1 + b2 + c4 \ vs \ a2 + b1 + c1$$
$$a1 + b2 + c3 \ vs \ a2 + b1 + c2$$
$$a1 + b2 + c3 \ vs \ a2 + b1 + c1$$
$$a1 + b2 + c2 \ vs \ a2 + b1 + c1$$

This process is performed for all matches. But, because the numbers of categories for the criteria are different, each match generates an idiosyncratic set of ambiguities that depends on the numbers of categories on the included criteria.

For example, the number of ambiguities generated from *abc vs abc* with 011 *vs* 100 (as above) is different to the number from *abd vs abd* with 011 *vs* 100 when the numbers of categories on criteria *c* and *d* are different. Therefore, in general the ambiguities for each of the $\frac{x!}{(x-z)!z!}$ bases (combinations of *z* criteria from the *x* criteria) must be generated individually.

For example, in the case of the base corresponding to the particular combination of z criteria comprising *the first* z of the x criteria (*criterion 1*, *criterion 2*, ... *criterion z*, where $z \leq x$), there are $2^{z-1}-1$ structures as before. (The choice of *the first* z of the x criteria, instead of any other z of the x criteria, is for notational simplicity.) Each structure has $\left(\frac{y_1(y_1-1)}{2}\right)\times\left(\frac{y_2(y_2-1)}{2}\right)\times\ldots\times\left(\frac{y_z(y_z-1)}{2}\right)$ ambiguities, where $y_1, y_2 \ldots y_z$ are the numbers of categories on each of these first z criteria.

The total number of z^{th}-degree ambiguities is obtained by summing the number of ambiguities (analogous to the example) across all $\frac{x!}{(x-z)!z!}$ bases. In general, the number of ambiguities of each degree can be calculated from the equation described below.

For a given APS with $y_1, y_2, \ldots y_x$ categories on the respective x criteria, the equation is based on the following definitions. First, Y is the set of the numbers of categories on the x criteria: $Y = \{y_1, y_2, \ldots y_x\}$. Second, C is the set of unordered z-tuples formed by taking all possible $\frac{x!}{(x-z)!z!}$ combinations of the elements of Y, z-at-a-time: $C = \{c \mid c$ is an unordered z-tuple from Y, as defined above$\}$. Each of C's elements (i.e., sets), c_i, is numbered from 1 to $\frac{x!}{(x-z)!z!}$: c_i, $i = 1, 2, \ldots \frac{x!}{(x-z)!z!}$. Each of c_i's elements, y_{ij}, is numbered from 1 to z: y_{ij}, $j = 1, 2, \ldots z$.

Applying these definitions, the number of unique ambiguities of a given degree (z), when the number of categories varies across criteria, is given by:

$$(2^{z-1}-1)\times\sum_{i=1}^{\frac{x!}{(x-z)!z!}}\prod_{j=1}^{z}\frac{y_{ij}(y_{ij}-1)}{2} \qquad (3)$$

Alternatively this equation can be expressed as: $(2^{z-1}-1)\times\sum_{c\in C}\prod_{y\in c}\frac{y(y-1)}{2}$.

For common values of y_{ij} ($y_{ij}=y$), that is, all criteria have the same numbers of categories, equation (3) is equivalent to equation (2) above.

Either process explained above — for the same or, alternatively, different numbers of categories on the criteria — can be used to generate all of an APS's ambiguities of a given degree (such as 3rd-degree, as above), or, alternatively, ambiguities can be generated individually.

As each ambiguity is generated it is tested for whether or not it is theoretically impossible, and therefore to be discarded or not, given the theoretically impossible profiles that were culled earlier (as described above). For the simple process for generating ambiguities described earlier, all theoretically impossible profiles are simply removed before ambiguities are generated from them.

However, because the 'efficient ambiguities generator' explained immediately above does not generate ambiguities from profiles, the ambiguities must be tested as they are generated for whether or not the profile pairs that they represent have been culled or not.

For example, for an APS with $x = 4$ and $y = 3$, the reduced form $a1 + b3$ *vs* $a3 + b2$ (i.e., 13__ *vs* 32__) represents nine ambiguously ranked profile pairs:

1311 *vs* 3211
1312 *vs* 3212
1313 *vs* 3213
1321 *vs* 3221
1322 *vs* 3222
1323 *vs* 3223
1331 *vs* 3231
1332 *vs* 3232
1333 *vs* 3233

For $a1 + b3$ *vs* $a3 + b2$ to be theoretically impossible and therefore discardable, in all nine of these pairs at least one of the profiles — a minimum of nine and a maximum of 18 profiles — must be theoretically impossible. If instead at least one of the nine pairs is not excluded, then $a1 + b3$ *vs* $a3 + b2$ is possible and therefore ought not to be discarded. Accordingly all nine pairs must be considered before it can be determined whether $a1 + b3$ *vs* $a3 + b2$ should be discarded or not, but as soon as one profile pair is found that has not been excluded, the ambiguity should be retained.

Enumerating all such profile pairs for any given ambiguity of degree z is relatively straight-forward, as each profile pair is based on the ambiguity in question, augmented by all possible combinations of the categories on the other $x - z$ criteria. There are therefore y^{x-z} such profile pairs when the number of categories on the criteria (y) is the same.

As noted earlier, the term y^{x-z} appears in equation (1) above — giving the *total* number of ambiguities (including replicates) of a given degree — but not in the otherwise identical equation (2) — giving the number of *unique* ambiguities (excluding replicates) of a given degree.

Accordingly y^{x-z} can be interpreted as the number of 'copies' of a particular ambiguity generated by the algorithmically simple process explained earlier. That is, in the example above with $x = 4$ and $y = 3$, each of the $y^{x-z} = 3^{4-2} = 9$ pairwise profile comparisons generates $a1 + b3$ *vs* $a3 + b2$ (of which eight are discarded because they are replicates).

Having generated the ambiguities, Step 2 of our method involves *explicitly* resolving them, one-at-a-time, while identifying all other ambiguities that are *implicitly* resolved as corollaries. This step is represented in Figure 5 at 520.

Any of the unresolved ambiguities of any degree could be selected for explicit resolution during the calibration process, however there tend to be fewer ambiguities to store at lower degrees, and lower degree ambiguities are easier for decision makers to resolve, and so the process starts by resolving the 2[nd]-degree ambiguities and then proceeds to

resolving successively higher-degree ambiguities. If instead it were desired that the number of decisions be minimised at the expense of the complexity of the decisions, the process should instead start by resolving the highest degree ambiguities and then proceed to resolving successively lower-degree ambiguities.

As ambiguities can only be resolved via value judgements, they must be decided by an individual 'decision maker' or group of 'decision makers', preferably with knowledge of the field in which the particular APS is to be applied. For example, a group of decision makers for a medical APS may comprise a panel of doctors and patients. Hereinafter decision makers (plural) are referred to.

Accordingly the preferences of the decision makers must be probed via a series of questions concerning their pairwise rankings of profiles.

As listed earlier, the ambiguities for the original exemplar APS with $x = 3$ and $y = 2$ are: (1) $b2 + c1$ *vs* $b1 + c2$, (2) $a2 + c1$ *vs* $a1 + c2$, (3) $a2 + b1$ *vs* $a1 + b2$, (4) $a2 + b2 + c1$ *vs* $a1 + b1 + c2$, (5) $a2 + b1 + c2$ *vs* $a1 + b2 + c1$ and (6) $a1 + b2 + c2$ *vs* $a2 + b1 + c1$.

For ambiguity (1) $b2 + c1$ *vs* $b1 + c2$, for example, the decision makers are asked, in essence: "Given two alternatives that are the same with respect to criterion *a*, which has the greater priority, _21 or _12?"

If there is more than one decision maker, the process of getting answers to this and subsequent questions can be streamlined by asking decision makers to cast votes (perhaps via email) for the pairwise rankings they favour.

However, in this type of situation the majority voting runs the risks of the well-known voting paradox, whereby, depending on the decision makers' *individual* rankings of the profiles, the order the ambiguities are voted on can determine the resolutions that are derived. To avoid this possibility, were it likely, the decision makers should instead be required to reach a consensus on their pairwise rankings.

Logically, three mutually exclusive and exhaustive answers to the above question are possible: (1) _21 is strictly preferred to _12 or (2) _12 is strictly preferred to _21 or (3) they are equally preferred (i.e., indifference between _21 and _12).

Notationally, these three preferences can be represented as (1) _21 > _12 or (2) _12 > _21 or (3) _21 = _12, corresponding to (1) $b2 + c1 > b1 + c2$ or (2) $b1 + c2 > b2 + c1$ or (3) $b2 + c1 = b1 + c2$ (where, as usual, ">" is "strictly greater than" and "=" is "equal to").

Weak preferences, for example, _21 is at least as preferred as _12 (notationally, _21 ≥ _12), are also a logical possibility. However strict preferences are more useful from a practical perspective.

The decision makers might, quite naturally, protest that their answer to the above question ("Given two alternatives that are the same with respect to criterion *a*, which has the greater priority, _21 or _12?") depends on whether criterion *a* is rated '1' or '2' (for both alternatives). Nonetheless, such distinctions are precluded by the internal logic of APSs. If the decision makers will not answer the question as it is posed then, in effect, the APS itself will answer it by default, since the ambiguity will eventually be implicitly resolved by the other explicitly resolved ambiguities chosen by the decision makers.

More specifically, for example, *if* the decision makers were to decide that 121 > 112 — corresponding to $b2 + c1 > b1 + c2$ — then this implies 221 > 212, and vice versa. And analogously for $b1 + c2 > b2 + c1$ and $b1 + c2 = b2 + c1$.

The key word above is "*if*", as clearly neither inequality holds intrinsically. Therefore a value judgement is required to resolve ambiguity (1) $b2 + c1$ *vs* $b1 + c2$: either 121 > 112 *and* 221 > 212 or 112 > 121 *and* 212 > 221 or 121 = 112 *and* 221 = 212. By virtue of the laws of arithmetic, if one inequality or equality holds then the other must too; if one does not then neither does the other. It is impossible to have one half of either proposition without the other.

Accordingly the question above could be rephrased, in essence, as: "Which one of the following three possible rankings of two alternatives do you prefer, (1) 121 > 112 *and* 221 > 212 or (2) 112 > 121 *and* 212 > 221 or (3) 121 = 112 *and* 221 = 212?"

Continuing with the example, suppose that in fact the decision makers resolve ambiguity (1) by choosing _21 > _12 (in other words 121 > 112 *and* 221 > 212), corresponding to $b2 + c1 > b1 + c2$. Two alternative, but equivalent, approaches are available for identifying the implicitly resolved ambiguities.

Although these approaches — hereinafter referred to as 'Approach 1' (of which there are two variants) and 'Approach 2' — differ in the means and the sequence in which the implicitly resolved ambiguities are identified, both generate the *same* list of explicitly resolved ambiguities, from which the point values are derived at Step 3 described below.

However because Approach 1 becomes relatively unwieldy, and therefore more resource intensive to implement, for more criteria and categories than the exemplar APS with $x = 3$ and $y = 2$, the particularly preferred embodiment of the computer program of the invention is based on Approach 2. Nonetheless, Approach 1 is described first as it is more intuitively tractable and is therefore useful for illustrating Approach 2.

Approach 1 may be summarised as follows. After a given ambiguity is explicitly resolved by the decision makers, *all* other ambiguities that are implicitly resolved as corollaries are immediately identified by adding appropriate inherent inequalities and/or other explicitly resolved ambiguities (inequalities or equalities). Then another (unresolved) ambiguity is explicitly resolved by the decision makers and all its corollaries are identified. The process is repeated until all ambiguities have been resolved, either explicitly or implicitly.

Thus a corollary of the *explicit* resolution of ambiguity (1) as $b2 + c1 > b1 + c2$ (decided by the decision makers, as explained earlier) is the *implicit* resolution of ambiguity (4) $a2$

$+ b2 + c1$ *vs* $a1 + b1 + c2$. This is revealed by *adding* the inherent inequality $a2 > a1$ and $b2 + c1 > b1 + c2$: $(a2 > a1) + (b2 + c1 > b1 + c2) = (a2 + b2 + c1 > a1 + b1 + c2)$.

Although this addition is mathematically legitimate, its theoretical validity in the context of APSs rests on the assumption that the decision makers are logically consistent in the sense that their pairwise profile rankings are transitive. Transitivity means in general that if alternatives $A > B$ and $B > C$, then $A > C$.

Thus, in the present example, the *explicit* resolution of ambiguity (1) as $b2 + c1 > b1 + c2$ corresponds to 121 > 112 and 221 > 212, as discussed earlier. Moreover 212 > 112 because $a2 > a1$. Therefore, assuming profile rankings are transitive, 221 > 212 and 212 > 112 implies 221 > 112 — corresponding to $a2 + b2 + c1 > a1 + b1 + c2$, as was revealed above by adding $a2 > a1$ and $b2 + c1 > b1 + c2$.

As there are no other corollaries at this point, the next ambiguity on the list, in this case ambiguity (2) $a2 + c1$ *vs* $a1 + c2$, can be presented to the decision makers to resolve via an analogous question to the first one: "Given two alternatives that are the same with respect to criterion b, which has the greater priority, 1_2 or 2_1?"

Suppose the decision makers answer 1_2 > 2_1, corresponding to $a1 + c2 > a2 + c1$. A corollary is the implicit resolution of ambiguity (6) $a1 + b2 + c2$ *vs* $a2 + b1 + c1$, as revealed by adding inherent inequality $b2 > b1$ to $a1 + c2 > a2 + c1$: $(b2 > b1) + (a1 + c2 > a2 + c1) = (a1 + b2 + c2 > a2 + b1 + c1)$.

In addition, both (1) $b2 + c1 > b1 + c2$ and (2) $a1 + c2 > a2 + c1$ implicitly resolve ambiguity (3) $a2 + b1$ *vs* $a1 + b2$, as revealed by their addition: $(b2 + c1 > b1 + c2) + (a1 + c2 > a2 + c1) = (a1 + b2 > a2 + b1)$ (after cancelling the c terms).

As for the addition of inherent inequalities and explicitly resolved ambiguities (as above), this addition is also justified by the assumption that the pairwise profile rankings of the decision makers are transitive.

Thus, in the present example, the *explicit* resolution of ambiguity (1) as $b2 + c1 > b1 + c2$ corresponds to 121 > 112 and 221 > 212, and (2) $a1 + c2 > a2 + c1$ corresponds to 112 > 211 and 122 > 221. Given 121 > 112 and 112 > 211, then, by transitivity, 121 > 211 is implied. Likewise, given 122 > 221 and 221 > 212, then 122 > 212 is implied. Both 121 > 211 and 122 > 212 correspond to $a1 + b2 > a2 + b1$, as was revealed above by adding (1) $b2 + c1 > b1 + c2$ and (2) $a1 + c2 > a2 + c1$.

Thus from just two explicit decisions to resolve ambiguities (1) and (2), another three ambiguities (3, 4 and 5) are implicitly resolved, so that five of the six ambiguities are resolved.

The remaining ambiguity, ambiguity (5) $a2 + b1 + c2$ *vs* $a1 + b2 + c1$, must be explicitly resolved by the decision makers, via a question that is conceptually simpler than the two earlier ones: "Which alternative has the greater priority, 212 or 121?" Suppose the decision makers answer 212 > 121, corresponding to $a2 + b1 + c2 > a1 + b2 + c1$.

The system is now fully specified as: (1) $b2 + c1 > b1 + c2$, (2) $a1 + c2 > a2 + c1$, (3) $a1 + b2 > a2 + b1$, (4) $a2 + b2 + c1 > a1 + b1 + c2$, (5) $a2 + b1 + c2 > a1 + b2 + c1$ and (6) $a1 + b2 + c2 > a2 + b1 + c1$ — as well as the inherent inequalities $a2 > a1$, $b2 > b1$ and $c2 > c1$. Of inequalities (1) to (6), only three (1, 2 and 5) were explicitly resolved by the decision makers, with the other three (3, 4 and 6) implicitly resolved as corollaries.

Finally, in general but not in the present example, any explicitly resolved ambiguities that are themselves corollaries of other explicitly resolved ambiguities can be removed from the list from which the point values are derived at Step 3 (below). This is because only independent inequalities/equalities are required for deriving point values.

A variant of the approach outlined above — in effect, its converse — is the identification of the *sufficient* (but not necessary) *conditions* for implicitly resolving ambiguities, in terms of (other) resolved ambiguities. Ambiguities are either explicitly resolved by the

decision makers or implicitly resolved when their sufficient conditions are met. Any ambiguities whose sufficient conditions are not met must therefore be resolved explicitly by the decision makers, until all ambiguities have been resolved, either explicitly or implicitly.

Thus in the exemplar APS with $x = 3$ and $y = 2$, ambiguities (4), (5) and (6) have four sufficient conditions each in terms of resolved 2nd-degree ambiguities. Specifically, ambiguity (4) is (implicitly) resolved as $a2 + b2 + c1 > a1 + b1 + c2$ if at least one of the following inequalities holds: $a2 + c1 > a1 + c2$ or $a2 + c1 = a1 + c2$ (in both cases because $b2 > b1$) or $b2 + c1 > b1 + c2$ or $b2 + c1 = b1 + c2$ (in both cases because $a2 > a1$).

These and analogous sufficient (but not necessary) conditions for ambiguities (5) and (6) to be implicitly resolved as $a2 + b1 + c2 > a1 + b2 + c1$ and $a1 + b2 + c2 > a2 + b1 + c1$ respectively are listed in Figure 10. No sufficient conditions exist for the opposite resolutions of the three ambiguities nor for equalities (that is, not for RHS > LHS nor LHS = RHS) in terms of resolved 2nd-degree ambiguities.

It is then simply a matter of comparing these conditions against resolved ambiguities (1), (2) and (3) (arrived at via Approach 1 explained above). Accordingly, as identified via shading in Figure 10, (1) $b2 + c1 > b1 + c2$ implicitly resolves ambiguity (4) as $a2 + b2 + c1 > a1 + b1 + c2$, and both (2) $a1 + c2 > a2 + c1$ and (3) $a1 + b2 > a2 + b1$ implicitly resolve ambiguity (6) as $a1 + b2 + c2 > a2 + b1 + c1$.

These are the same 3rd-degree resolutions as were revealed earlier via the explicit resolutions of ambiguities (1) and (2) and their additions to inherent inequalities and each other respectively. As such they reflect, as before, pairwise profile rankings that are transitive.

Finally, as before, ambiguity (5) remains to be explicitly resolved, and then the system is fully specified.

This variant of Approach 1 generalises for APSs with higher-degree ambiguities. Their sufficient conditions are in terms of both individual lower-degree resolved ambiguities and combinations of them, of which there may be many. For example, the sufficient conditions for the resolution of a 5^{th}-degree ambiguity are in terms of combinations of 2^{nd}-degree resolved ambiguities, 3^{rd}-degree resolved ambiguities and 4^{th}-degree resolved ambiguities.

However, sufficient conditions for an ambiguity of a given degree can also be identified in terms of resolved ambiguities of the same, or even higher, degrees. In the present example, sufficient conditions can also be identified for resolving 2^{nd}-degree ambiguities in terms of other resolved 2^{nd}-degree ambiguities. For example, a sufficient condition for resolving ambiguity (3) $a2 + b1$ *vs* $a1 + b2$ as $a1 + b2 > a2 + b1$ is (1) $b2 + c1 > b1 + c2$ *and* (2) $a1 + c2 > a2 + c1$, corresponding to $(b2 + c1 > b1 + c2) + (a1 + c2 > a2 + c1)$, as explained earlier.

This means that in general it is difficult to enumerate and check *all* possible sufficient conditions (to ensure that none are missed); therefore in practice, this variant must be supplemented by other methods, such as the first variant of Approach 1 explained above.

Finally, with respect to both variants of Approach 1, because ambiguities (1) to (3) are not independent, both the order and the manner in which they are resolved affects the number of explicit value judgements that are required. The maximum number required is four and the minimum is two.

For example, if inequality (3) had been decided *before* inequalities (1) and (2) (instead of after, as above), then all three ambiguities (as well as ambiguity 5) would have had to have been resolved explicitly. This is because inequality (3) $a1 + b2 > a2 + b1$ cannot be added to (1) $b2 + c1 > b1 + c2$ or (2) $a1 + c2 > a2 + c1$ to obtain the other inequality, and yet inequalities (1) and (2) imply (3). For ambiguities (4) to (6), on the other hand, one and only one must be resolved explicitly.

Alternatively, for example, had ambiguity (1) been resolved as $b2 + c1 = b1 + c2$ instead of $b2 + c1 > b1 + c2$ (indifference rather than strict preference), and ambiguity (2) resolved as $a1 + c2 > a2 + c1$ (as above), then no other explicit resolutions would have been necessary.

This can be confirmed by noting that (1) $b2 + c1 = b1 + c2$ implies both (4) $a2 + b2 + c1 > a1 + b1 + c2$ and (5) $a2 + b1 + c2 > a1 + b2 + c1$. In other words $(b2 + c1 = b1 + c2) + (b2 > b1)$ for both of them and (2) $a1 + c2 > a2 + c1$ implies (6) $a1 + b2 + c2 > a2 + b1 + c1$ (as above) and $(b2 + c1 = b1 + c2) + (a1 + c2 > a2 + c1)$ implies (3) $a1 + b2 > a2 + b1$. Clearly, the point values derived from this system of equations and inequalities would be different to the point values derived earlier.

Similarly the system would be completely specified by the explicit resolution of ambiguities (1) and (6) as $b1 + c2 > b2 + c1$ and $a2 + b1 + c1 > a1 + b2 + c2$ only. Of particular interest is: $(a2 + b1 + c1 > a1 + b2 + c2) + (b2 > b1) =$ (2) $(a2 + c1 > a1 + c2)$; and $(a2 + b1 + c1 > a1 + b2 + c2) + (c2 > c1) =$ (3) $(a2 + b1 > a1 + b2)$. This illustrates the fact that it is not necessary to resolve lower-degree (here 2[nd]-degree) ambiguities *before* implicitly resolving higher- degree (here 3[rd]-degree) ambiguities. The process can be reversed, as illustrated here.

Unfortunately, as mentioned earlier, both variants of Approach 1 become unwieldy for APSs with more criteria and categories than the exemplar APS with $x = 3$ and $y = 2$. This is because calculating and managing *all* possible combinations of additions or possible sufficient conditions is resource intensive.

The 'additions' variant of Approach 1 involves maintaining a list of all possible inequalities/equalities that result from the addition of each explicitly resolved ambiguity with each other explicitly resolved ambiguity as well as with the inherent inequalities, and with every sum generated, and each of these sums with each other recursively until all possible additive combinations are exhausted. Each newly generated ambiguity is

'checked off' against the list of implicitly resolved ambiguities: if it is not on the list then it is yet to be resolved.

Similarly, the 'sufficient conditions' variant of Approach 1 involves managing a list of all implicitly resolved ambiguities, identifying all possible sufficient conditions of lower degrees, of which there may be a great many combinations, and identifying ambiguities implied by the same or higher degree explicitly resolved ambiguities (by some other means).

Approach 2, on the other hand, which is explained below, is more efficient and is therefore the preferred method used in the computer program of the invention.

In summary, Approach 2 involves testing ambiguities individually for whether or not they are implicitly resolved as corollaries of the explicitly resolved ambiguities (that were resolved earlier). If a given ambiguity is identified as having been implicitly resolved then it is deleted. If instead it is not implicitly resolved then it must be explicitly resolved by the decision makers. The process is repeated until all ambiguities have been identified as having been implicitly resolved or they are explicitly resolved.

Thus, with reference to the exemplar APS with $x = 3$ and $y = 2$, after the decision makers resolve ambiguity (1) as $b2 + c1 > b1 + c2$ as described earlier, the next ambiguity on the list ((2) $a1 + c2$ *vs* $a2 + c1$) is tested for whether or not it is implicitly resolved as a corollary of $b2 + c1 > b1 + c2$, as well as the inherent inequalities $a2 > a1$, $b2 > b1$ and $c2 > c1$.

Here, as in the computer program of the invention, this and subsequent tests can be performed via linear programming. In effect, this test is performed by asking the following two hypothetical questions (of the method, not the decision makers).

Hypothetical Question 1: *If* it were the case that the ambiguity in question [here (2) $a1 + c2$ *vs* $a2 + c1$] had been implicitly resolved as LHS > RHS (i.e., $a1 + c2 > a2 + c1$), then

does a solution exist to the system comprising this hypothetical inequality and the (actual) explicitly resolved inequalities/equalities [here (1) $b2 + c1 > b1 + c2$] — as well as the inherent inequalities [here $a2 > a1$, $b2 > b1$ and $c2 > c1$]? (*Yes* or *No*?)

If the answer is *No* — and therefore it would not be theoretically possible for the decision makers, if they wanted to, to explicitly resolve the ambiguity in question as LHS > RHS (here $a1 + c2 > a2 + c1$) — then it must be true that either RHS > LHS or LHS = RHS (i.e., either $a2 + c1 > a1 + c2$ or $a1 + c2 = a2 + c1$). This implies that the ambiguity in question (here ambiguity 2) has been implicitly resolved (i.e., as either $a2 + c1 > a1 + c2$ or $a1 + c2 = a2 + c1$). Hence it is of no further use and can be deleted.

If instead the answer to Question 1 is *Yes* — and therefore it would be theoretically possible for the decision makers, if they wanted to, to explicitly decide LHS > RHS (here $a1 + c2 > a2 + c1$) — then the following second hypothetical question is asked.

Hypothetical Question 2: *If* it were instead the case that the ambiguity in question (here ambiguity 2) had been implicitly resolved as RHS > LHS (that is, $a2 + c1 > a1 + c2$), then does a solution exist to the system comprising this hypothetical inequality and the (actual) explicitly resolved inequalities/equalities [here (1) $b2 + c1 > b1 + c2$, as before] — as well as the inherent inequalities [here $a2 > a1$, $b2 > b1$ and $c2 > c1$]? (*Yes* or *No*?)

If the answer is *No* — and therefore it would not be theoretically possible for the decision makers, if they wanted to, to explicitly resolve the ambiguity in question as RHS > LHS (here $a2 + c1 > a1 + c2$) — then it must be true that either LHS > RHS or LHS = RHS (either $a1 + c2 > a2 + c1$ or $a1 + c2 = a2 + c1$). This implies that the ambiguity in question (here ambiguity 2) has been implicitly resolved, in this case as $a1 + c2 > a2 + c1$ or $a1 + c2 = a2 + c1$. Hence it is of no further use and can be discarded.

If instead the answer to Question 2 is *Yes* then it must be inferred that as well as it being theoretically possible for the decision makers, if they wanted to, to explicitly resolve the ambiguity in question as LHS > RHS (here $a1 + c2 > a2 + c1$) from Question 1 that it is

also theoretically possible for them to resolve it as RHS > LHS ($a2 + c1 > a1 + c2$). This implies that the ambiguity in question (here ambiguity 2) is *not* implicitly resolved as a corollary of the explicitly resolved ambiguities, and therefore it must be explicitly resolved by the decision makers.

In the case of ambiguity (2) $a1 + c2$ *vs* $a2 + c1$ the answers to Questions 1 and 2 are *Yes* and *Yes*, and so the ambiguity should be presented to the decision makers for them to explicitly resolve. As for the earlier demonstration of Approach 1, suppose it is decided $a1 + c2 > a2 + c1$.

The next ambiguity on the list — (3) $a2 + b1$ *vs* $a1 + b2$ — is then tested via the same process as outlined above. This time, though, the list of (actual) explicitly resolved inequalities/equalities against which (3) $a2 + b1 > a1 + b2$ (i.e., LHS > RHS, as for Question 1) and then, if necessary, (3) $a1 + b2 > a2 + b1$ (RHS > LHS, as for Question 2) are tested comprises (2) $a1 + c2 > a2 + c1$ as well as (1) $b2 + c1 > b1 + c2$ (as before).

Thus the list of explicitly resolved inequalities/equalities is continually updated — including in general but not in the present example, as for Approach 1, the identification of any explicitly resolved inequalities/equalities that are themselves corollaries of others on the list.

The process is repeated for all ambiguities until all of them have been identified as having been implicitly resolved or they are explicitly resolved by the decision makers. This can be summarised for the present example as follows.

For ambiguity (3) the answer to Question 1 is *No*, and therefore this ambiguity is identified as having been implicitly resolved as a corollary of the explicitly resolved ambiguities. For ambiguity (4), the answer to Question 1 is *Yes* but the answer to Question 2 is *No*, and therefore this ambiguity is identified as having been implicitly resolved.

For ambiguity (5) the answers to both questions are *Yes*, implying that this ambiguity has not been implicitly resolved, and so it must be explicitly resolved by the decision makers: as $a2 + b1 + c2 > a1 + b2 + c1$, as for the demonstration of Approach 1. Finally, for ambiguity (6) the answers are *Yes* and *No*, and therefore this ambiguity is also identified as having been implicitly resolved.

Thus, of the six ambiguities, three had to be explicitly resolved by the decision makers — (1) $b2 + c1 > b1 + c2$, (2) $a1 + c2 > a2 + c1$ and (5) $a2 + b1 + c2 > a1 + b2 + c1$ (the same three as for Approach 1) — with the other three (3, 4 and 6) identified as having been implicitly resolved as corollaries.

The final step of the method (Step 3) of the invention involves simultaneously solving the system of (independent) explicitly resolved ambiguities (inequalities and equalities) and inherent inequalities to obtain the point values. This step of the method is represented in Figure 5 at 530.

As described above, any explicitly resolved ambiguities that are themselves corollaries of other explicitly decided inequalities are removed, because only independent inequalities/equalities are required for deriving the point values.

For the exemplar APS with $x = 3$ and $y = 2$, one solution for (1) $b2 + c1 > b1 + c2$, (2) $a1 + c2 > a2 + c1$ and (5) $a2 + b1 + c2 > a1 + b2 + c1$ and $a2 > a1$, $b2 > b1$ and $c2 > c1$ is: $a1 = 0$, $a2 = 2$, $b1 = 0$, $b2 = 4$, $c1 = 0$ and $c2 = 3$. These point values produce ranking #9 in Figure 4.

The method explained above may be summarised as follows. First (Step 1), generate the ambiguities of the APS that is to be calibrated as shown at 510 in Figure 5.

Then (Step 2) *explicitly* resolve them via the value judgements of the consulted decision makers, while identifying all other ambiguities that are *implicitly* resolved as corollaries, until all ambiguities are resolved. This step is illustrated at 520 in Figure 5. Although, in

theory, the implicitly resolved ambiguities can be identified via *Approach 1* or *Approach 2* (both explained above), the latter is more efficient and therefore it is the particularly preferred method for the computer program of the invention, as explained below.

Finally (Step 3), simultaneously solve the system of explicitly resolved ambiguities (that is, inequalities and equalities) and inherent inequalities to obtain the point values for the APS. This step is illustrated at 530 in Figure 5.

The property (assumption) that the profile rankings of the decision makers are transitive (or logically consistent), enables the number of ambiguities that must be explicitly resolved at Step 2 to be minimised, with the remainder emerging implicitly as corollaries.

For APSs with more criteria and categories than the exemplar with $x = 3$ and $y = 2$, minimising the number of explicitly resolved ambiguities, and accordingly the number of value judgements required from the decision makers, is a significant practical advantage of our method.

The computer program for implementing the method described above will now be described more particularly. The computer program of the invention comprises, in broad terms, five steps, as explained in turn below. An overview of the five steps is illustrated in Figure 11.

Before starting the program, the user — who may be one of the decision makers consulted to resolve the ambiguities, or, alternatively, he or she may be a facilitator of the calibration process — must have chosen the criteria and categories of the APS that is to be calibrated and ranked each criterion's categories.

At Step 1, when the program begins, the user may be asked to enter a title for the APS and the criteria and their categories, and to rank the categories for each criterion. The criteria and categories must be labelled in terms of the variables (e.g. '$a1$', '$a2$' '$a3$', etc.) and verbally described in preparation for their later presentation to the decision makers.

The user is given the opportunity of listing theoretically impossible combinations of criteria and categories that partially or fully specify profiles — known as the *To Be Excluded* list — and that are therefore to be used to cull ambiguities that are generated by the program. This list may be left empty if the user wishes.

After the program is initialised, it calculates the number of unique ambiguities to be resolved using equation (3) above. This can be reported to the user and used to estimate whether the system can be solved in an acceptable amount of time given the computing resources that are available.

Step 2 of the program involves generating the ambiguities using the 'efficient ambiguities generator' described earlier. An overview of this step is illustrated in Figure 12.

For simplicity and efficiency, ambiguities are generated one degree at a time, beginning with the 2nd-degree, as determined by the value of a control variable — known as the *Current Degree* variable — which is initially set to 2.

As each ambiguity is generated it is checked for whether or not it is theoretically possible, given the *To Be Excluded* list of partially or fully specified profiles. If the ambiguity is theoretically possible it is then tested via Approach 2 of Step 2 of the method of the invention explained earlier for whether or not it is implicitly resolved by the explicitly resolved ambiguities. An overview of the procedure is illustrated in Figure 13.

This and similar tests may be performed via linear programming, which is described below.

If an ambiguity is found to be implicitly resolved then it is discarded; otherwise it is added to a list known as the *To Be Resolved* list. (Note that when the 2nd-degree

ambiguities are generated, none will be implicitly resolved because none have yet been explicitly resolved.)

Step 3 of the program is to present the *To Be Resolved* list of unresolved ambiguities for the current degree to the user. An overview of this step and Step 4 (explained below) is illustrated in Figure 14.

The user can choose to view the ambiguities either in equation form (for example, $a1 + b2$ *vs* $a4 + b1$) or symbolically (for example, 21_ _ _ or 41_ _ _), which are ordered either randomly or by their criteria and categories. The user is invited to select an ambiguity for the purpose of explicitly resolving it. The selected ambiguity is 'translated' verbally in terms of the criteria and category descriptions.

If the user desires, he or she may skip this particular ambiguity and select another one, or she may resolve it by clicking one of three buttons labelled (in essence): "LHS greater" or "RHS greater" (i.e., < or RHS preferred) or "LHS and RHS equal". If "RHS greater (preferred)" is chosen, for consistency, the LHS and RHS of the resolved ambiguity are switched and stored as RHS > LHS.

The system can also permit weak inequalities (such as "LHS greater than or equal to" and "RHS greater than or equal to"), however this means that some ambiguities will later be partially (weakly) solved and so some buttons in the user interface must be disabled when the ambiguity is selected. Note that in any resulting APS the result will be either "greater than" or it will be "equal to" but it will not be both. The system can also be designed so that only strong inequalities and no equalities are permitted, thereby producing strict profile rankings only.

Step 4 is for the program to remove the explicitly resolved ambiguity (as above) from the *To Be Resolved* list and add it — as either an inequality or equality (depending on how the ambiguity was resolved) — to the list of explicitly resolved ambiguities, known as the *Explicitly Resolved* list. An overview of this step and Step 3 is illustrated as Figure 14.

Each inequality/equality on the *Explicitly Resolved* list is then tested as to whether or not it is implied by the others on the list. Any found to be implied may be marked as being 'redundant' and hereinafter ignored, but not deleted because they may be re-used later if any explicitly resolved ambiguities are later revised by the decision makers (explained below).

All ambiguities on the *To Be Resolved* list are then tested for whether or not their resolution is implied by the (non-redundant) inequalities/equalities on the *Explicitly Resolved* list, and if so they are deleted from the *To Be Resolved* list. This test was explained in earlier in terms of Hypothetical Questions 1 and 2 of Approach 2 of the method's Step 2 and is illustrated in Figure 13.

In essence, the test involves finding whether or not a solution (in terms of feasible point values) exists to a system comprising the explicitly resolved inequalities/equalities and inherent inequalities, and each of the possible hypothetical inequalities in turn (that is LHS > RHS and RHS > LHS) corresponding to the ambiguity in question. This and the earlier tests based on determining the existence of solutions are performed via linear programming (LP) with inequality and equality constraints. LP is discussed in more detail in the final section below.

If the *To Be Resolved* list is not empty, the program returns to Step 3. If instead the *To Be Resolved* list is empty and the *Current Degree* is equal to the number of criteria (x) in the particular APS being calibrated, the program proceeds to Step 5. Alternatively, if the *Current Degree* is less than the APS's number of criteria (and the *To Be Resolved* list is empty), the *Current Degree* is increased by '1' and the program returns to Step 2.

The user may choose at any time to view the explicitly resolved ambiguities for all degrees and 'undo' any of them. Explicitly resolved ambiguities that are 'redundant' are displayed in a different colour to alert the user to the fact that undoing them alone will

have no material effect. An overview of the 'undo module', as referred to in Figure 11, is illustrated in Figure 15.

When explicitly resolved ambiguities are undone, they are deleted from the *Explicitly Resolved* list. If any non-redundant explicitly resolved ambiguities are undone, the explicitly resolved ambiguities marked redundant are re-tested for whether or not they are still redundant, and marked accordingly.

Also, if non-redundant ambiguities are undone, the *Current Degree* variable is reset to the degree of the lowest-degree ambiguity that was undone, and the program restarts from the second step in the computer program as set out above.

When all the ambiguities for all degrees have been resolved, the list of explicitly resolved inequalities/equalities and inherent inequalities is solved via linear programming for the point values of the APS. These are presented to users, as well as a range of 'summary statistics', including the numbers of ambiguities and explicitly resolved ambiguities at each degree and, it the user wants them, the inequalities/equalities that were chosen.

In the interests of deriving point values that are integers and as low as possible, thereby maximising their 'user-friendliness', integer programming and an objective function that minimises the sum of the variables (for example, $a1 + a2 + a3 + b1 + b2 + b3 + c1 + c2 + c3$, etc.) may be specified.

Alternatively, were it desired by users that the point values be normalised to a particular range for the profile total scores, such as 0 to 100, the derived point values are scaled accordingly. However this usually forces some point values to be fractions, which is not as 'user-friendly' as integers. Accordingly both integer and normalised values can be calculated and presented.

As explained above, linear programming (LP) may be used in the computer program of the invention. Specifically, LP may be used at Steps 2 and 4 to determine whether the

resolution of an ambiguity is implied by the inequalities/equalities on the *Explicitly Resolved* list. In Step 5, LP is used for deriving the point values when all ambiguities have been resolved. The following features must be observed when LP is used.

First, the inequalities/equalities and inherent inequalities must be converted to a form suitable for LP. For example, the inherent inequalities $a3 > a2 > a1$ must be written as $a2 - a1 = 1$ and $a3 - a2 = 1$, and the explicitly resolved ambiguities $a2 + b2 = a3 + b1$ and $a2 + b3 > a3 + b2$ as $a2 + b2 - a3 - b1 = 0$ and $a2 + b3 - a3 - b2 = 1$. Setting the RHS of the weak inequality to "1" (that is, 'epsilon') corresponds to the initial inequality being strict, although other values for epsilon may perform at least as well.

Second, in Steps 2 and 4, the LP objective function need only be "0", as all that is being tested for is the *existence* of a solution rather than a particular optimal solution.

Finally, because the variables corresponding to the lowest category on the respective criteria ($a1$, $b1$, $c1$, and so on) are, effectively, 'numeraires' or 'baseline' values for the respective criteria, they can be set equal to zero, thereby eliminating them from the system to be solved and increasing the efficiency of the LP algorithm.

Below is an illustration of how LP may be used to test whether or not an ambiguity on the *To Be Resolved* list is implied by the inequalities/equalities on the *Explicitly Resolved* list.

For the sake of the example, suppose the ambiguity in question is $a1 + b3 + c3$ *vs* $a2 + b1 + c1$ and the following inequalities/equalities are on the *Explicitly Resolved* list (where the shaded inequality signifies that it is 'redundant' in the sense, as discussed earlier, that it is implied by others on the list).

$$\begin{gathered} a2 + b2 = a3 + b1 \\ a2 + b3 > a3 + b2 \\ a3 + c1 > a1 + c3 \\ a2 + c1 > a1 + c3 \\ b1 + c2 = b3 + c1 \\ b1 + c3 = b3 + c2 \\ a1 + b2 + c3 > a2 + b1 + c1 \end{gathered}$$

In addition, the inherent inequalities of this exemplar APS with $x = 3$ and $y = 3$ are:

$$a3 > a2 > a1$$
$$b3 > b2 > b1$$
$$c3 > c2 > c1$$

These inequalities/equalities (and ignoring the shaded redundant inequality) — and given $a1 = b1 = c1 = 0$ (as discussed above) — are represented in a form suitable for LP, as follows. By definition, a solution (in terms of the point values) exists to the LP problem:

$$\begin{aligned}
&\text{minimise } 0\\
\text{subject to: } & a2 + b2 - a3 = 0\\
& a2 + b3 - a3 - b2 \geq 1\\
& a2 - c3 \geq 1\\
& c2 - b3 = 0\\
& c3 - b3 - c2 = 0\\
& b2 + c3 - a2 \geq 1\\
& a2 \geq 1\\
& a3 - a2 \geq 1\\
& b2 \geq 1\\
& b3 - b2 \geq 1\\
& c2 \geq 1\\
& c3 - c2 \geq 1
\end{aligned}$$

To test whether $a1 + b3 + c3$ *vs* $a2 + b1 + c1$ is implied by the inequalities/equalities on the *Explicitly Resolved* list, the above LP problem is first augmented with $b3 + c3 - a2 \geq 1$ (i.e., $a1 + b3 + c3 > a2 + b1 + c1$), and tested for whether or not a solution to this new problem exists.

A solution does exist, and so the original LP problem is next augmented with $a2 - b3 - c3 \geq 1$ (i.e., $a2 + b1 + c1 > a1 + b3 + c3$) and tested for whether or not a solution to this second new problem exists.

In this case there is no solution, and so it must be inferred that either $a1 + b3 + c3 > a2 + b1 + c1$ or $a1 + b3 + c3 = a2 + b1 + c1$ is implied by the inequalities/equalities on the *Explicitly Resolved* list. The first inequality above is easily confirmed here by adding $b3 > b2$ to the last explicitly resolved ambiguity on the first list above: $(b3 > b2) + (a1 + b2$

$+ c3 > a2 + b1 + c1) = (a1 + b3 + c3 > a2 + b1 + c1)$. This alternative approach is variant 1 of Approach 1 explained earlier.

Therefore the ambiguity is implicitly resolved and deleted from the *To Be Resolved* list. (This process is performed for all the ambiguities on the *To Be Resolved* list.)

The foregoing describes the invention including preferred forms thereof. Alterations and modifications as will be obvious to those skilled in the art are intended to be incorporated within the scope hereof.

Paul Hanson and Franz Omble

By the authorised agents

AJ PARK

Per

Melissa Gill

Western Canada Waiting List Project

HIP AND KNEE REPLACEMENT PRIORITY CRITERIA

PLEASE PRINT CLEARLY

Provincial Name: ____________________

Patient Age: __________ Sex: [circle one] M F

[Tick one box] ☐ Left Hip ☐ Right Hip ☐ Left Knee ☐ Right Knee

[Tick one box] ☐ Primary ☐ Revision

Diagnosis: ____________________

____

Surgeon's Name: ____________________ Phone: ____________________

Date of Patient Evaluation: ____________________

Patients must be on appropriate non-surgical treatment prior to evaluation (e.g. medications, walking aids, shoe inserts)

PLEASE CHECK THE BOX THAT MOST ACCURATELY DESCRIBES THE PATIENT'S CURRENT SITUATION

1. Pain on motion (e.g. walking, bending):*
   - ☐ None/Mild
   - ☐ Moderate
   - ☐ Severe
2. Pain at rest (e.g. while sitting, lying down or causing sleep disturbance):*
   - ☐ None
   - ☐ Mild
   - ☐ Moderate
   - ☐ Severe

*Take into account usual duration, intensity, and frequency of pain, including need for narcotic vs. non-narcotic medication

3. Ability to walk:
   - ☐ Over 5 blocks
   - ☐ 1-5 blocks
   - ☐ <1 blocks
   - ☐ Household ambulator
4. Other functional limitations (e.g. putting on shoes, managing stairs, sitting to standing, sexual activity, bathing, cooking, recreation or hobbies):
   - ☐ No limitations
   - ☐ Mild limitations (able to do most activities with minor modifications or difficulty)
   - ☐ Moderate limitations (able to do most activities but with modification or assistance)
   - ☐ Severe limitations (unable to perform most activities)

FIGURE 1

5. Abnormal findings on physical exam related to affected joint (e.g. deformity, instability, leg length difference, restriction of range of motion on examination):
   - ☐ None/Mild
   - ☐ Moderate
   - ☐ Severe

6. Potential for progression of disease documented by radiographic findings (e.g. recurrent dislocation, x-ray evidence of protrusion, significant bone loss, component wear, impending fracture):**
   - ☐ None
   - ☐ Mild
   - ☐ Moderate
   - ☐ Severe

** Predominantly applies to revisions, use in primary cases only in special circumstances (e.g. ligament instability, bone loss)

7. Threat to patient role and independence in society (e.g. ability to work, give care to dependents, live independently (difficulty must be related to affected joint)):
   - ☐ Not threatened but more difficult
   - ☐ Threatened but not immediately
   - ☐ Immediately threatened or unable

8. All things considered, how would you rate the urgency or relative priority of this patient? (Draw a line across the scale)

   |————————————————————————|

   Not Urgent at all — Extremely Urgent (just short of an emergency)

9. In your clinical judgement, what should be the maximum waiting time for this patient?
   Number of weeks ____________ OR Number of months ____________

10. In your practice how long would it take this patient to have surgery done from the time you first see the patient?
    Number of days ________ OR Number of weeks ________ OR Number of months ________

Please record any comments on the form, criteria or process: ____________________

____

____

____

____

____

FIGURE 2

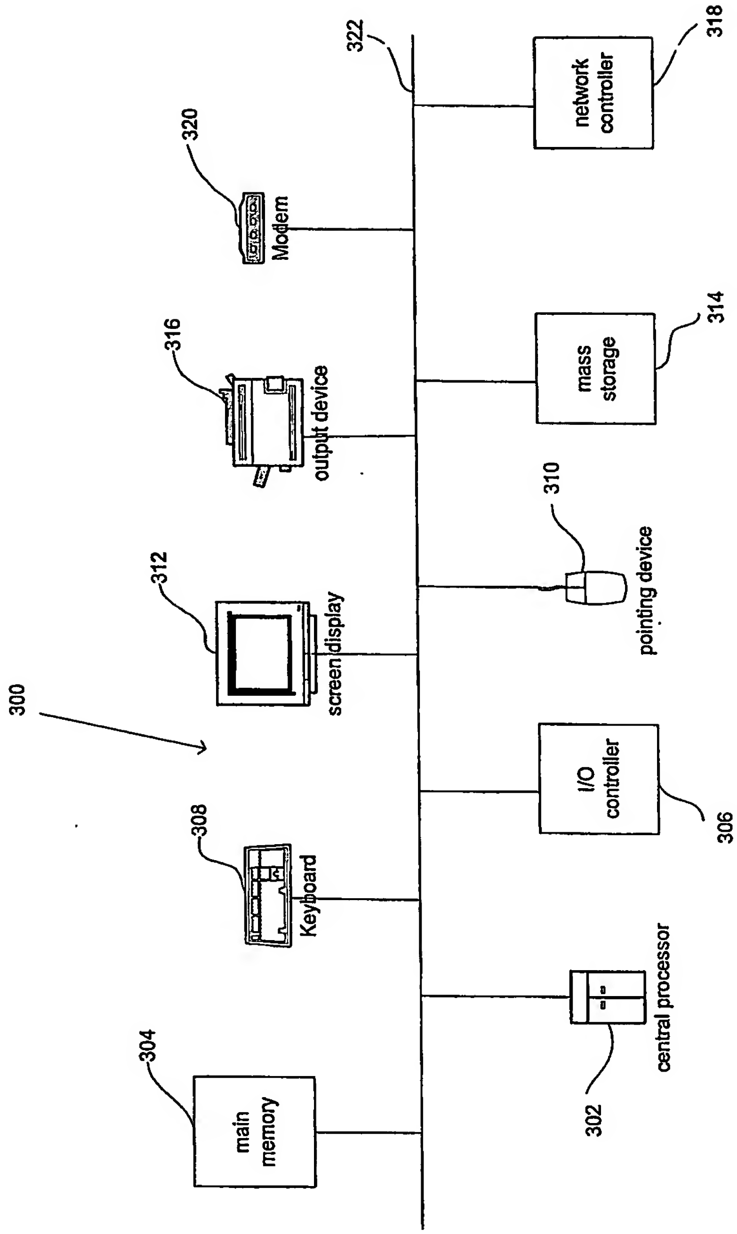
300
main memory
304
central processor
302
Keyboard
308
I/O controller
306
screen display
312
pointing device
310
output device
316
mass storage
314
Modem
320
network controller
318
322

FIGURE 3

```
                 + 211 - 122 - 121 - 112 - 111     (Ranking #1)
             +212
             |   + 122 - 211 - 121 - 112 - 111     (Ranking #2)
     + 221|
     |       |   + 212 - 121 - 211 - 112 - 111     (Ranking #3)
     |       +122
     |           + 121 - 212 - 211 - 112 - 111     (Ranking #4)
     |
     |           + 211 - 122 - 112 - 121 - 111     (Ranking #5)
     |       +221
     |       |   + 122 - 211 - 112 - 121 - 111     (Ranking #6)
222 + 212|
     |       |   + 221 - 112 - 211 - 121 - 111     (Ranking #7)
     |       +122
     |           + 112 - 221 - 211 - 121 - 111     (Ranking #8)
     |
     |           + 212 - 121 - 112 - 211 - 111     (Ranking #9)
     |       +221
     |       |   + 121 - 212 - 112 - 211 - 111     (Ranking #10)
     + 122|
             |   + 221 - 112 - 121 - 211 - 111     (Ranking #11)
             +212
                 + 112 - 221 - 121 - 211 - 111     (Ranking #12)
```

FIGURE 4

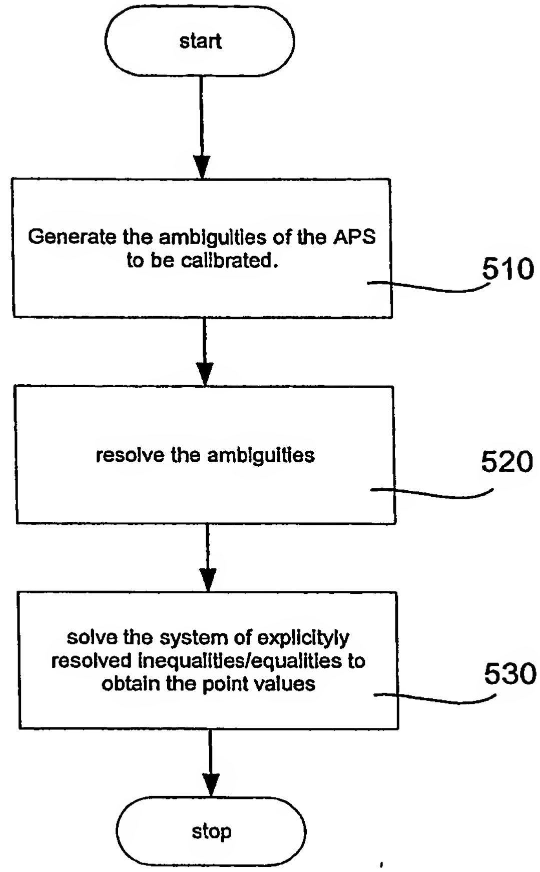
start
Generate the ambiguities of the APS to be calibrated.
510
resolve the ambiguities
520
solve the system of explicityly resolved inequalities/equalities to obtain the point values
530
stop

FIGURE 5

versus (*vs*)	221 $a2+b2+c1$	212 $a2+b1+c2$	122 $a1+b2+c2$	112 $a1+b1+c2$	121 $a1+b2+c1$	211 $a2+b1+c1$
221 $a2+b2+c1$		$b2+c1$ *vs* $b1+c2$	$a2+c1$ *vs* $a1+c2$	$a2+b2+c1$ *vs* $a1+b1+c2$	n.a.	n.a.
212 $a2+b1+c2$			$a2+b1$ *vs* $a1+b2$	n.a.	$a2+b1+c2$ *vs* $a1+b2+c1$	n.a.
122 $a1+b2+c2$				n.a.	n.a.	$a1+b2+c2$ *vs* $a2+b1+c1$
112 $a1+b1+c2$					$b1+c2$ *vs* $b2+c1$	$a1+c2$ *vs* $a2+c1$
121 $a1+b2+c1$						$a1+b2$ *vs* $a2+b1$
211 $a2+b1+c1$						

FIGURE 6

Criteria (x)	Categories (y)	Profiles (y^x)	Pairwise comparisons $y^x(y^x-1)/2$	Degrees	Total ambiguities (Equation 1)	Unique ambiguities (Equation 2)
3	2	8	28	2nd	6	3
				3rd	3	3
				All degrees:	*9*	*6*
3	3	27	351	2nd	81	27
				3rd	81	81
				All degrees:	*162*	*108*
3	4	64	2,016	2nd	432	108
				3rd	648	648
				All degrees:	*1,080*	*756*
4	2	16	120	2nd	24	6
				3rd	24	12
				4th	7	7
				All degrees:	*55*	*25*
4	3	81	3,240	2nd	486	54
				3rd	972	324
				4th	567	567
				All degrees:	*2,025*	*945*
4	4	256	32,640	2nd	3,456	216
				3rd	10,368	2,592
				4th	9,072	9,072
				All degrees:	*22,896*	*11,880*
4	5	625	195,000	2nd	15,000	600
				3rd	60,000	12,000
				4th	70,000	70,000
				All degrees:	*145,000*	*82,600*
5	2	32	496	2nd	80	10
				3rd	120	30
				4th	70	35
				5th	15	15
				All degrees:	*285*	*90*
5	3	243	29,403	2nd	2,430	90
				3rd	7,290	810
				4th	8,505	2,835
				5th	3,645	3,645
				All degrees:	*21,870*	*7,380*

FIGURE 7

Criteria (x)	Categories (y)	Profiles (y^x)	Pairwise comparisons $y^x(y^x-1)/2$	Degrees	Total ambiguities (Equation 1)	Unique ambiguities (Equation 2)
5	4	1,024	523,776	2nd	23,040	360
				3rd	103,680	6,480
				4th	181,440	45,360
				5th	116,640	116640
				All degrees:	*424,800*	*168,840*
5	5	3,125	4,881,250	2nd	125,000	1,000
				3rd	750,000	30,000
				4th	1,750,000	350,000
				5th	1,500,000	1,500,000
				All degrees:	*4,125,000*	*1,881,000*
6	2	64	2,016	2nd	240	15
				3rd	480	60
				4th	420	105
				5th	180	90
				6th	31	31
				All degrees:	*1,351*	*301*
6	3	729	265,356	2nd	10,935	135
				3rd	43,740	1,620
				4th	76,545	8,505
				5th	65,610	21,870
				6th	22,599	22,599
				All degrees:	*219,429*	*54,729*
6	4	4,096	8,386,560	2nd	138,240	540
				3rd	829,440	12,960
				4th	2,177,280	136,080
				5th	2,799,360	699,840
				6th	1,446,336	1,446,336
				All degrees:	*7,390,656*	*2,295,756*
10	4	1,048,576	549,755,289,60	2nd	106,168,320	1,620
				3rd	1,274,019,840	77,760
				4th	7,803,371,520	1,905,120
				5th	30,098,718,72	29,393,280
				6th	77,755,023,36	303,730,560
				7th	135,444,234,24	2,116,316,160
				8th	153,584,087,04	9,599,005,440
				9th	102,792,499,20	25,698,124,80
				10th	30,898,215,93	30,898,215,93
				All degrees:	*539,756,33876*	*68,646,770,67*

FIGURE 8

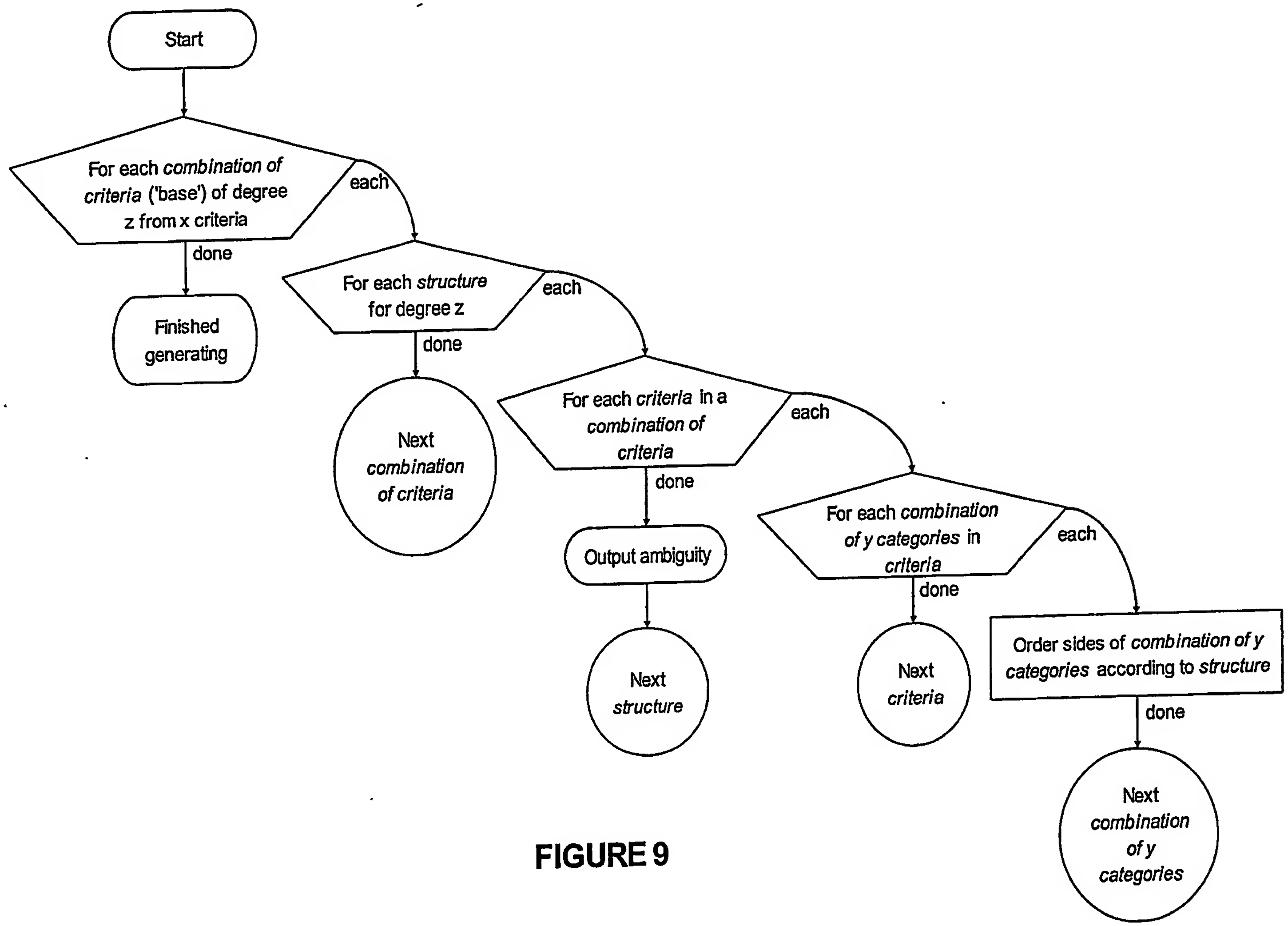
Start
For each *combination of criteria* ('base') of degree z from x criteria
each
done
Finished generating
For each *structure* for degree z
each
done
Next *combination of criteria*
For each *criteria* in a *combination of criteria*
each
done
Output ambiguity
Next *structure*
For each *combination of y categories* in *criteria*
each
done
Next *criteria*
Order sides of *combination of y categories* according to *structure*
done
Next *combination of y categories*

FIGURE 9

3^{rd}-degree ambiguity	Sufficient conditions for LHS > RHS of the 3^{rd}-degree ambiguity
(4) $a2 + b2 + c1$ *vs* $a1 + b1 + c2$	$a2 + c1 > a1 + c2$ or $a2 + c1 = a1 + c2$ or $b2 + c1 > b1 + c2$ or $b2 + c1 = b1 + c2$
(5) $a2 + b1 + c2$ *vs* $a1 + b2 + c1$	$b1 + c2 > b2 + c1$ or $b1 + c2 > b2 + c1$ or $a2 + b1 = a1 + b2$ or $a2 + b1 = a1 + b2$
(6) $a1 + b2 + c2$ *vs* $a2 + b1 + c1$	$a1 + c2 > a2 + c1$ or $a1 + c2 = a2 + c1$ or $a1 + b2 > a2 + b1$ or $a1 + b2 = a2 + b1$

FIGURE 10

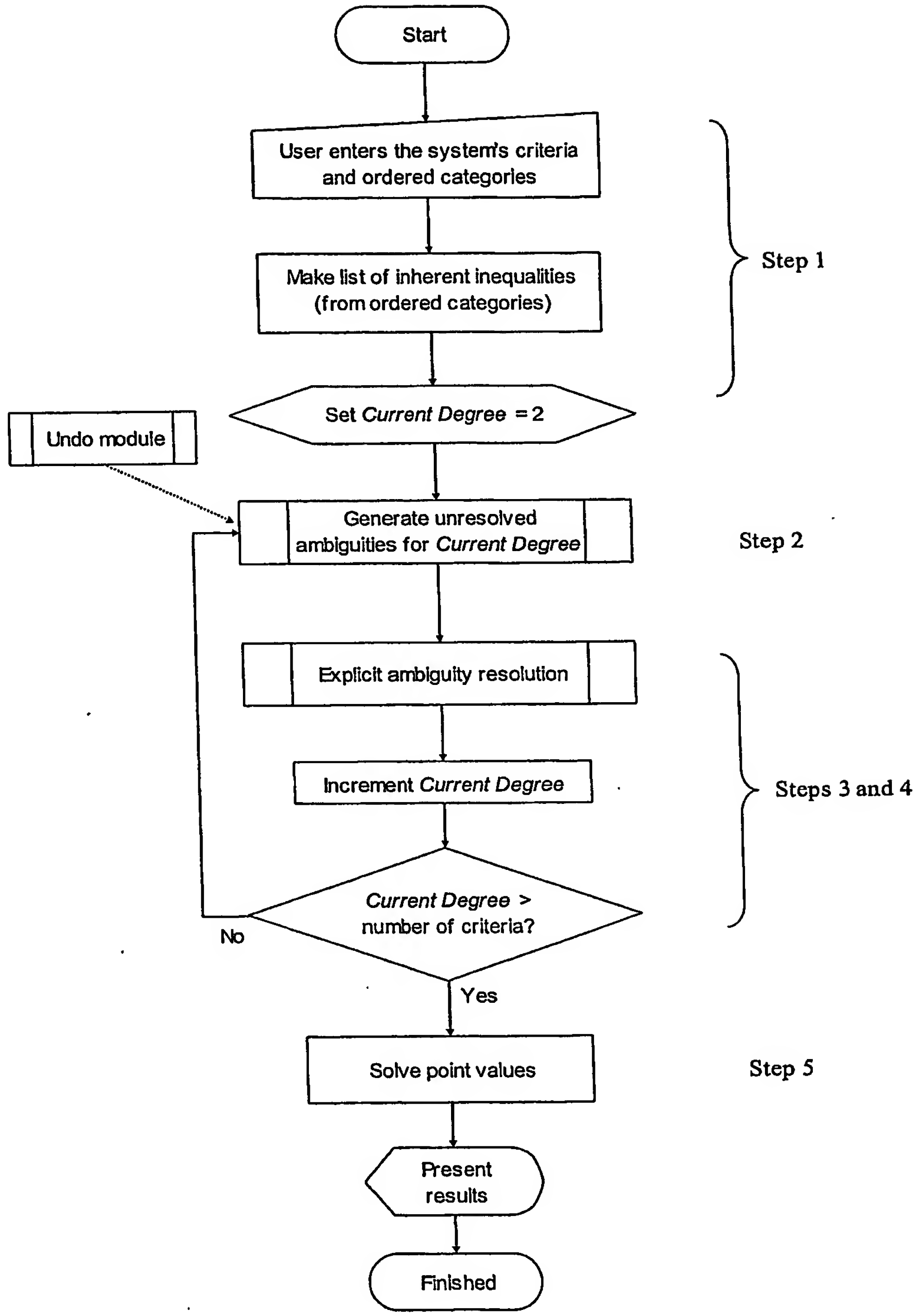
Start
User enters the system's criteria and ordered categories
Make list of inherent inequalities (from ordered categories)
Step 1
Set *Current Degree* = 2
Undo module
Generate unresolved ambiguities for *Current Degree*
Step 2
Explicit ambiguity resolution
Increment *Current Degree*
Steps 3 and 4
Current Degree > number of criteria?
No
Yes
Solve point values
Step 5
Present results
Finished

FIGURE 11

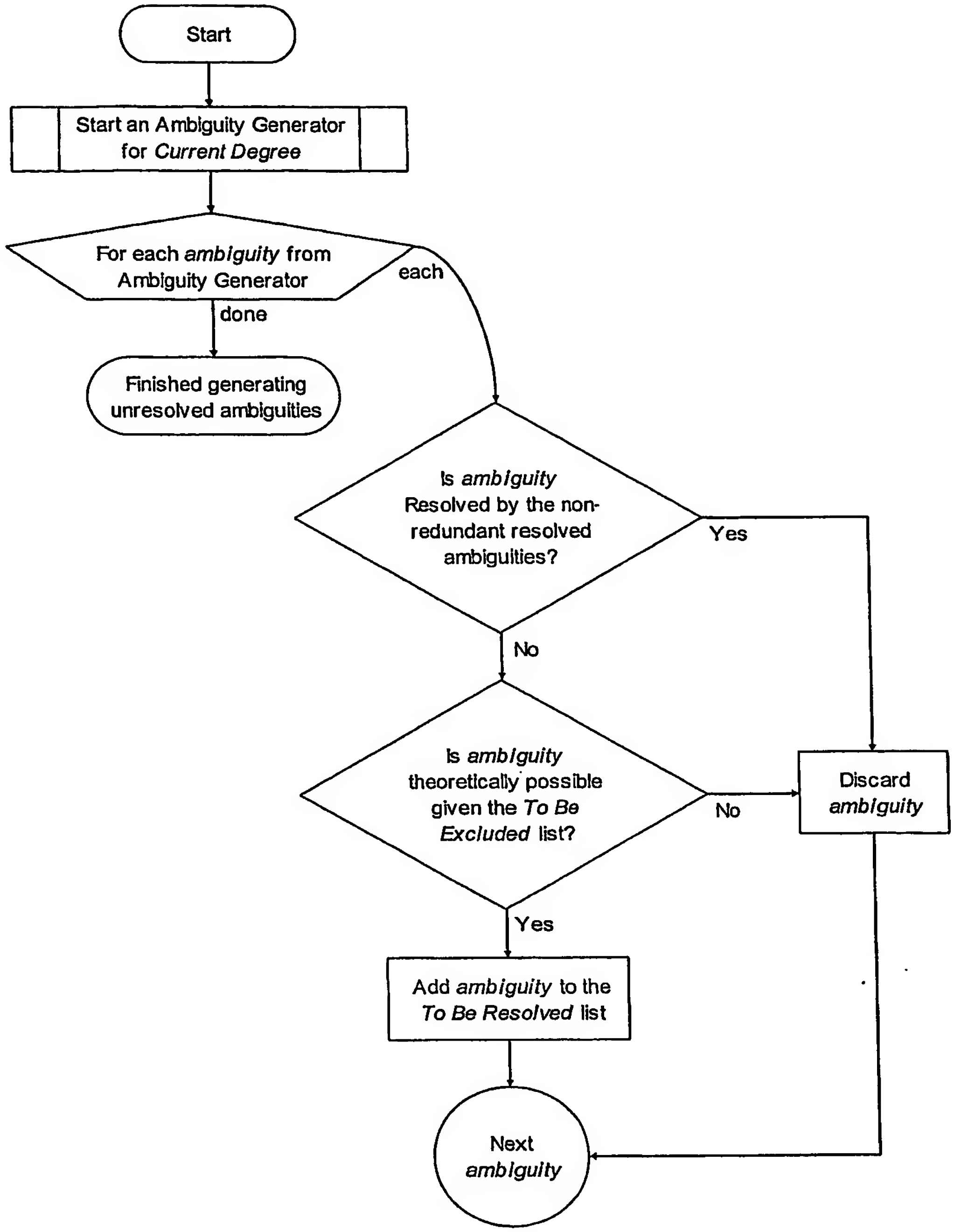
Start
Start an Ambiguity Generator for *Current Degree*
For each *ambiguity* from Ambiguity Generator
each
done
Finished generating unresolved ambiguities
Is *ambiguity* Resolved by the non-redundant resolved ambiguities?
Yes
No
Is *ambiguity* theoretically possible given the *To Be Excluded* list?
No
Discard *ambiguity*
Yes
Add *ambiguity* to the *To Be Resolved* list
Next *ambiguity*

FIGURE 12

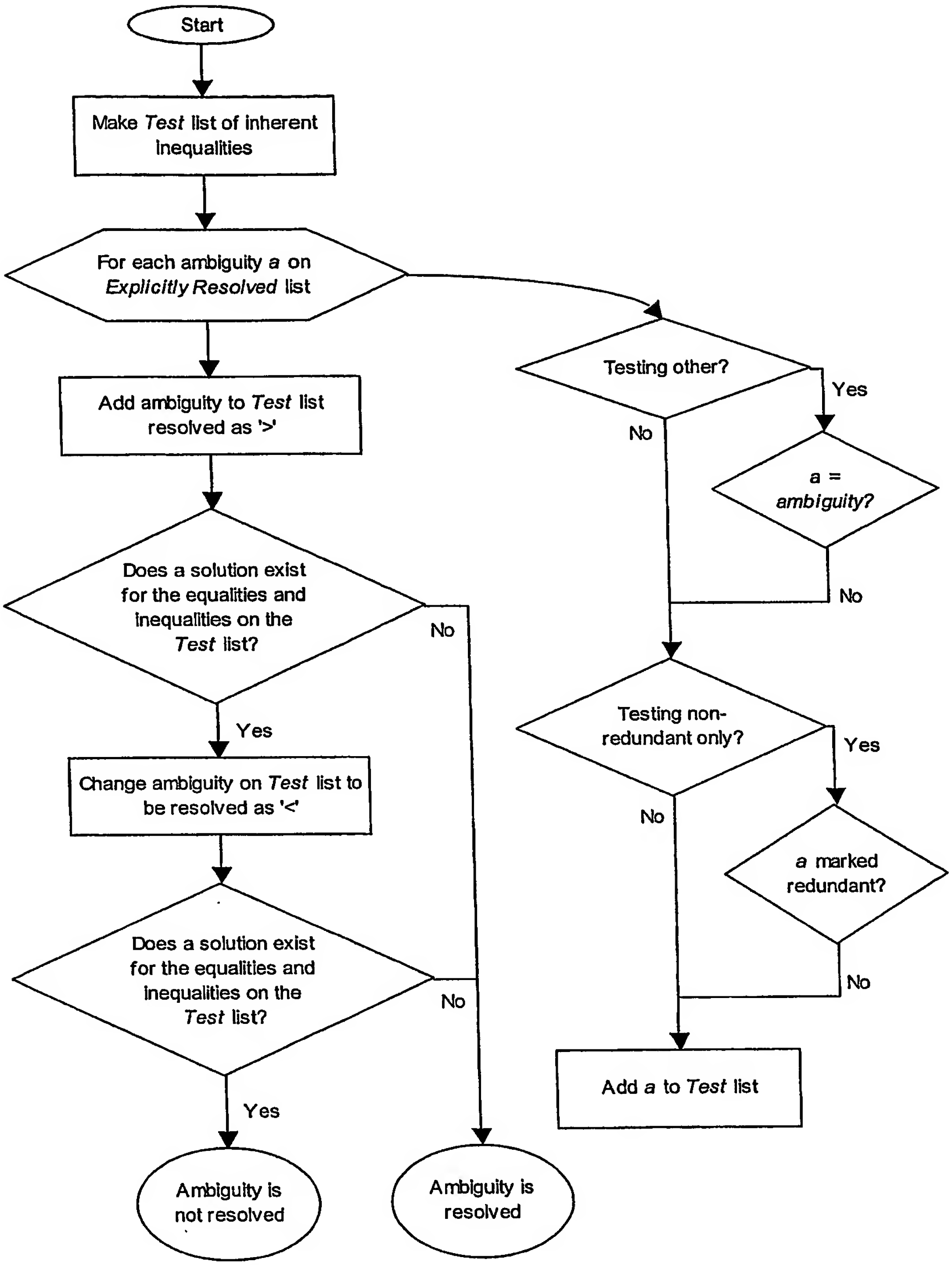
Start
Make Test list of inherent inequalities
For each ambiguity a on Explicitly Resolved list
Add ambiguity to Test list resolved as '>'
Does a solution exist for the equalities and inequalities on the Test list?
No
Yes
Change ambiguity on Test list to be resolved as '<'
Does a solution exist for the equalities and inequalities on the Test list?
No
Yes
Ambiguity is not resolved
Ambiguity is resolved
Testing other?
Yes
No
a = ambiguity?
No
Testing non-redundant only?
Yes
No
a marked redundant?
No
Add a to Test list

FIGURE 13

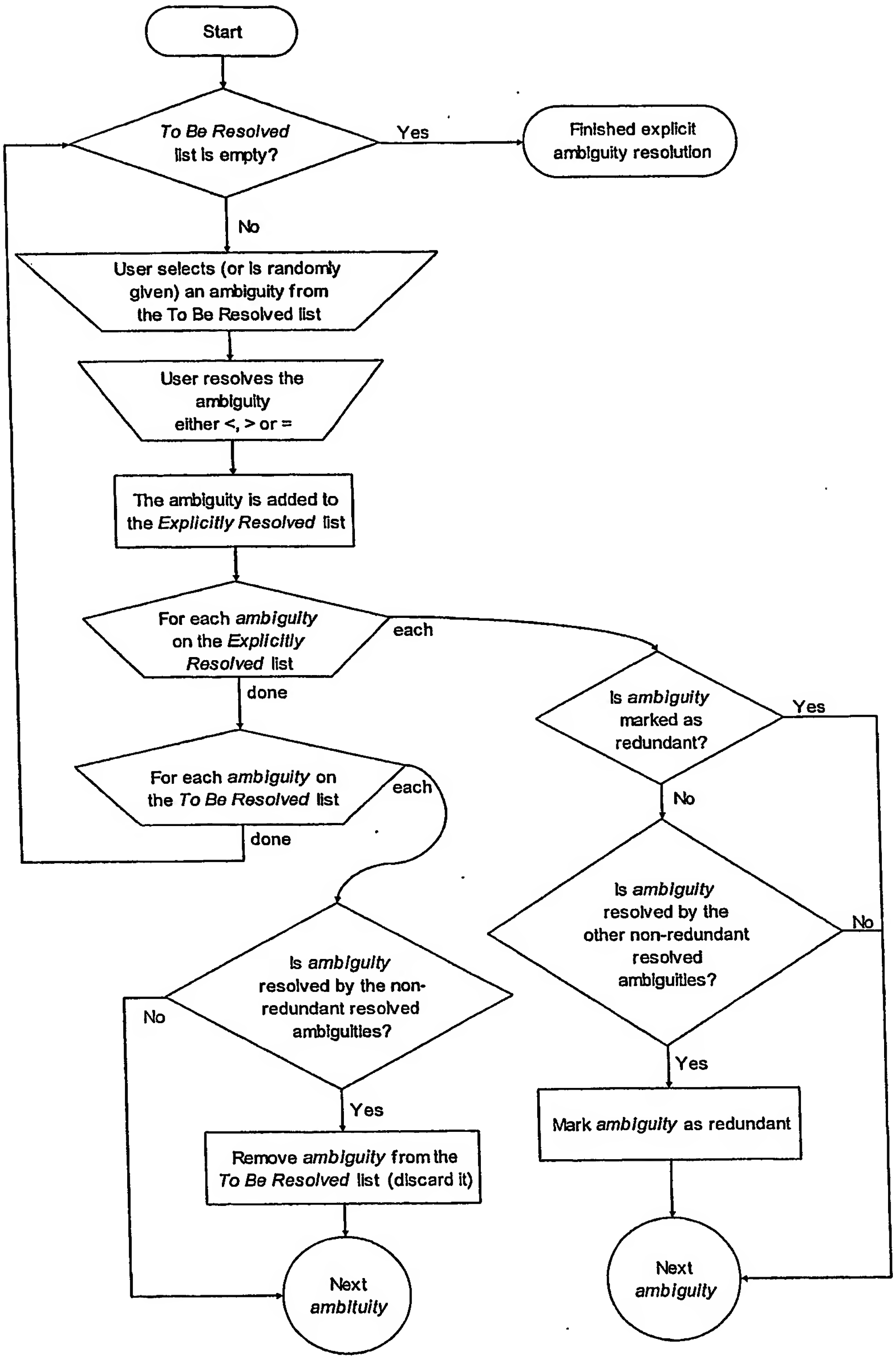
Start
To Be Resolved list is empty?
Yes
Finished explicit ambiguity resolution
No
User selects (or is randomly given) an ambiguity from the To Be Resolved list
User resolves the ambiguity either <, > or =
The ambiguity is added to the Explicitly Resolved list
For each ambiguity on the Explicitly Resolved list
each
done
Is ambiguity marked as redundant?
Yes
No
For each ambiguity on the To Be Resolved list
each
done
Is ambiguity resolved by the other non-redundant resolved ambiguities?
No
Yes
Is ambiguity resolved by the non-redundant resolved ambiguities?
No
Yes
Mark ambiguity as redundant
Remove ambiguity from the To Be Resolved list (discard it)
Next ambituity
Next ambiguity

FIGURE 14

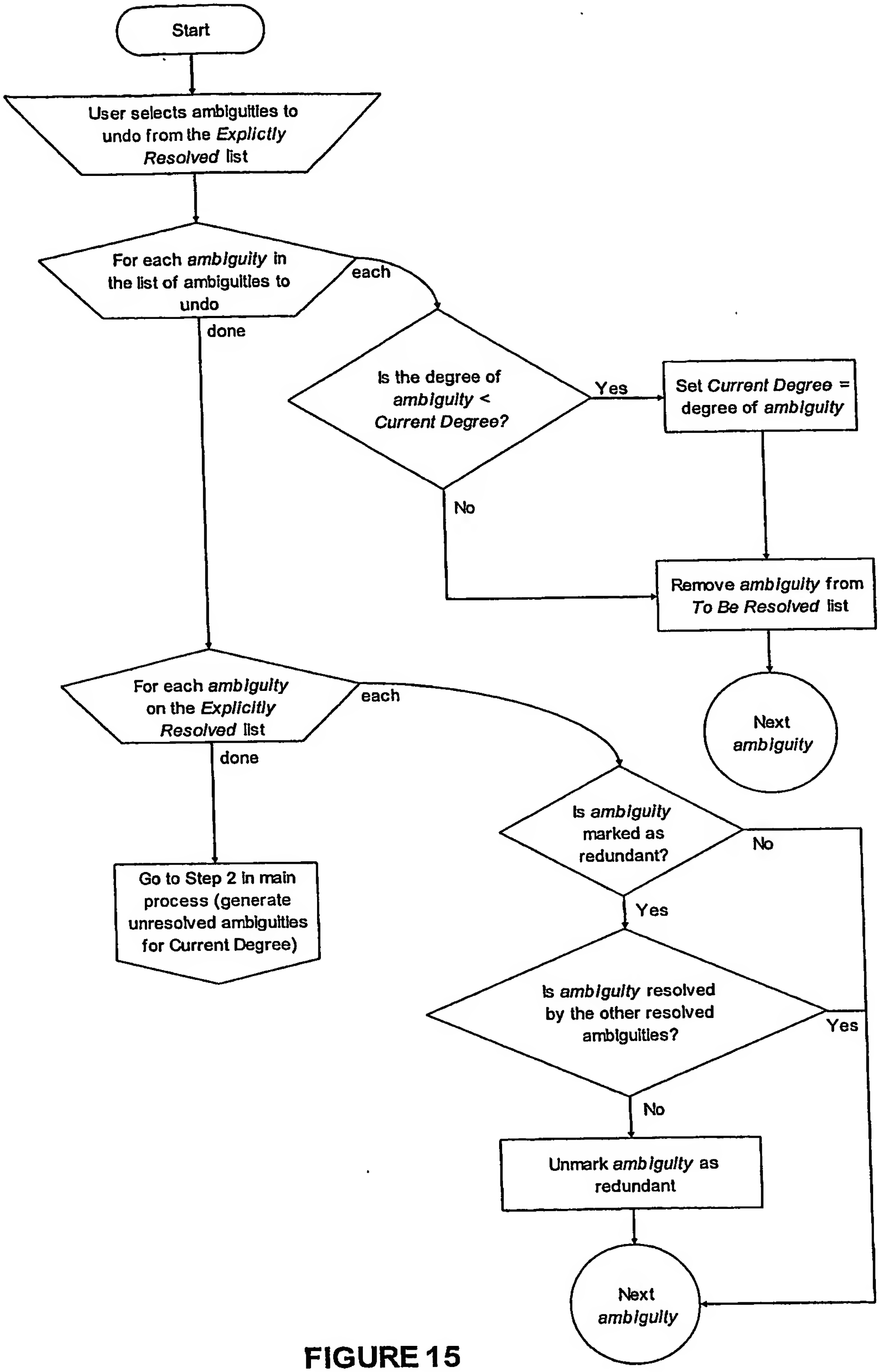
Start
User selects ambiguities to undo from the *Explictly Resolved* list
For each *ambiguity* in the list of ambiguities to undo
each
done
Is the degree of *ambiguity* < *Current Degree?*
Yes
No
Set *Current Degree* = degree of *ambiguity*
Remove *ambiguity* from *To Be Resolved* list
Next *ambiguity*
For each *ambiguity* on the *Explicitly Resolved* list
each
done
Is *ambiguity* marked as redundant?
No
Yes
Go to Step 2 in main process (generate unresolved ambiguities for Current Degree)
Is *ambiguity* resolved by the other resolved ambiguities?
Yes
No
Unmark *ambiguity* as redundant
Next *ambiguity*

FIGURE 15